DOCUMENT RESUME

ED 093 294 IR 000 776

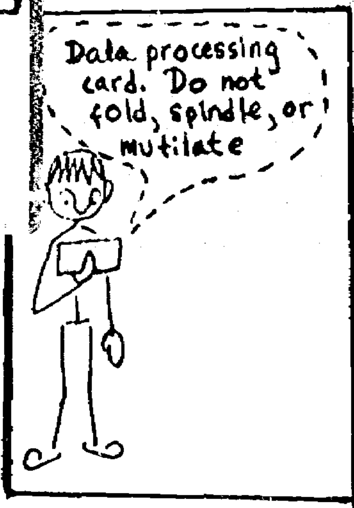
TITLE	Computers in Education Resource Handbook. Second Printing (With Revisions and Additions).
INSTITUTION	Oregon Univ., Eugene. Dept. of Computer Science.
PUB DATE	Jun 74
NOTE	508p.
AVAILABLE FROM	Department of Computer Science, University of Oregon, Eugene, Oregon 97403 ($10.00 plus $0.50 shipping and handling)
EDRS PRICE	MF-$0.90 HC-$24.60 PLUS POSTAGE
DESCRIPTORS	*Computer Oriented Programs; Computer Programs; *Computers; Computer Science; *Computer Science Education; Data Processing; *Education; *Educational Equipment; Educational Technology; Electronic Data Processing; Programing; Resource Guides; Time Sharing
IDENTIFIERS	University of Oregon

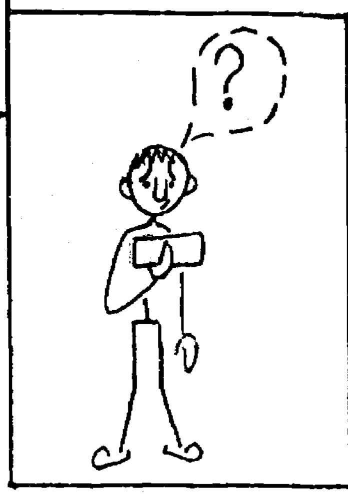
ABSTRACT

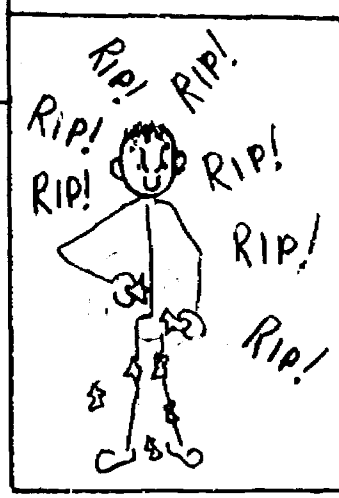
A lack of comprehensive up-to-date textbooks designed specifically for use in the computers in the education field motivated the development of this resource handbook. There are major sections entitled: computers in education; teaching about computers; the computer as an aid to learning; the computer as a teacher; the computer as a classroom management tool; administrative uses of computers; computing facilities; computers in special education; computers in research; and inservice training. Each chapter begins with an introductory overview and concludes with a section of abstracts of relevant articles. There are extensive lists of additional information sources, including books, periodicals, nonprint materials, people, and institutions. (WDR)

Suggested Price $10.00

COMPUTERS IN EDUCATION RESOURCE HANDBOOK







Dept. of Computer Science
University of Oregon
Eugene, Oregon 97403

First Printing, Sept. 1973
Second Printing (with revisions and additions)
June 1974

PREFACE

Although the field of computers in education has been growing rapidly for many years, it is still in its infancy. Especially at the public school level (grades K-12) little has been done relative to what will occur in the next two decades. Relatively few public school teachers or administrators are comfortable with computers, little good curriculum materials exist, and few schools have adequate plans for orderly exploitation of their growing computer capability.

Because of the size of the computers in education field, and its rapid growth, it is nearly impossible for one individual to encompass the area. The idea underlying this Handbook is that a number of people, working closely together and following a coordinated plan, can complete a task too large for a single individual. The writers of this Handbook are teachers and administrators who are involved in studying, teaching, and using computers in education. Their total knowledge and experience far exceeds that of any single individual. With minor exceptions the entire contents of this Handbook were written during Summer, 1973.

The ultimate goal of this Handbook is to improve the quality of education available to students. The Handbook itself contains some sections directed just at school administrators, some sections directed at teachers, and a number of sections of general interest to the student of the field of computers in education. Chapter I gives a more detailed discussion of the purpose of this Handbook, and contains abstracts of each section. It is expected that the reader will use the Table of Contents and Chapter I to find the sections of the Handbook he may wish to study in detail. Most of the Handbook presupposes some knowledge of the computer field. The reader with little or no knowledge of computers should read Chapter II before delving into subsequent parts of the book.

It is expected that this Handbook will be substantially revised during the coming year, and probably republished during 1974. Readers with suggestions for corrections, additions, or deletions should contact David Moursund, Dept. of Computer Science, University of Oregon, Eugene, Oregon 97403.

September 1973

The first printing of the Handbook sold out sooner than expected. It was therefore decided to do a modest amount of revision and to produce a second edition sooner than originally planned. Some material, deemed to be of lesser value, has been deleted in order to allow the addition of a substantial amount of new material. Of particular interest is the addition of Chapter 9: Computers in Special Education. Substantial changes have also been made to Chapter 3.

June 1974

ACKNOWLEDGEMENTS

This Handbook was conceived during conversations between Mike Dunlap and David Moursund in the fall of 1972. It is a logical outgrowth of curriculum materials development projects in a number of National Science Foundation supported institutes in computer science that have been run by Moursund, and a Computers in Education year long course taught by Dunlap during 1972-73. Much of the early planning for this Handbook was carried out by Dunlap and Moursund, assisted by various members of the Curriculum Taskgroup of the Oregon Council for Computer Education (OCCE). OCCE also helped to underwrite part of the costs involved in preparation of the manuscript.

Each contributor to the Handbook is an educator interested in applications of computer technology to the field of education. A number of the contributors have had many years of experience in using computers in education. To a large extent each section of the Handbook has a single author, or a small number of authors. In that case the authors are identified at the beginning of the section. A substantial amount of the work to produce the Handbook was carried out by educators participating in the course CS 507 (Computers in the Curriculum) given during the 8 week 1973 Summer Session at the University of Oregon. The writing of materials for the Handbook was a major activity of this course.

The following people each made substantial contributions to this Handbook:

Ed Anderson--Ed led the group working on the ADMINISTRATIVE USES OF COMPUTERS chapter, and served as editor for that chapter.

Cliff Burns--Cliff spelled out the format to be used for abstracts of related publications and supervised the preparation and classification of these abstracts. He also contributed to several sections of the Handbook.

Curtis Cook--Curtis served as co-leader and co-editor for the chapter on TEACHING ABOUT COMPUTERS.

Mike Dunlap--Mike contributed substantially in the planning, writing, and editing of the entire Handbook.

David Moursund--Dave contributed substantially in the planning and writing of this Handbook, and served as overall editor.

Jan Moursund--Jan served as copy editor for a large part of the Handbook (especially the Abstracts), and co-authored one section.

The typing of many rough drafts plus the typing of the entire final copy of this Handbook was carried out by Barbara Demezas and Barbara Platz during August 13, 1973 to September 12, 1973. This effort wore out two dozen IBM carbon ribbons, wore down twenty fingers, and wore thin the (previously) good working relations between David Moursund and the two Barbaras.

CONTENTS

**Note that Section I.B consists of brief abstracts of each section of this Handbook.

CHAPTER I INTRODUCTION

SECTION A: Purpose and Rationale of the Handbook

The computer is an exceedingly valuable tool in the field of education; its misuse, however, may contribute to a decrease in the quality of education in the United States. The computer will be used properly or misused, depending primarily upon the knowledge teachers and school administrators have in the area of computers in education. The purpose of this Handbook is to aid teachers and school administrators to increase their knowledge and understanding of the role and nature of computers in education.

Much of the early research and development work on computers was carried out by faculty members at various universities. Thus computers have been used in education since their first production. By and large, however, computers have entered most schools as money-saving administrative tools, for processing student records, payroll, etc. By the early 1960's a significant proportion of the larger colleges and universities were using computers for administrative purposes, and computer programming courses were beginning to enter their curricula. The 1960's saw considerable experimentation with instructional use of computers at the secondary, and even elementary, school levels. The computer as a media device (computer assisted instruction, computer aided learning) received widespread attention.

The late 1960's and early 1970's saw the establishment of hundreds of Computer Science Departments in colleges and universities, with degree programs at the associate, bachelor, master, and doctorate levels. The teaching of computer programming and the use of computers in mathematics courses became commonplace in many secondary schools. It was also during this time that some schools of higher education began to develop special programs of study for students interested in the computers in education field. For example, the Illinois Institute of Technology and the University of Oregon each developed master's degree programs in computer science education. A number of schools began to offer special teacher-oriented courses for pre-service teachers and for teachers returning to school for a master's or doctorate degree.

The University of Oregon began to offer courses specifically designed for education-oriented students during 1971, and has continued to do so on a regular basis. During 1973-74 two distinct levels of year long courses will be offered. One has no computer programming prerequisite, and the other is a graduate seminar for students with a good background in both computers and education. In teaching these courses it has become evident that there is a lack of up-to-date textbooks specifically designed to fit the needs of educators. This fact served as a prime motivation for the development of this Handbook.

At the current time (1973) the state of Oregon is one of the leaders in the instructional use of computers at the grades 1-12 level. For example, roughly half of the secondary schools in the state, representing roughly 3/4 of the student population at that level, have access to computers for instructional purposes. The METCOM system in the Portland area, the OTIS system in Lane County and seven other counties, and the Rogue Valley network all provide time-shared computing on Hewlett-Packard systems. A number of other schools and/or school districts have mini computers. Thus, for example, each of the four high schools in Salem has a mini computer and also has access to a time-shared system. Other modes of computer access used in Oregon include batch processing at school district offices, computing by mail, time-sharing on university computer networks, batch processing on college computers, use of computer time on business computers, and use of the Oregon Museum of Science and Industry computing facilities.

A large number of Oregon teachers have acquired substantial experience in the instructional use of computers under all kinds of conditions. Some of the knowledge and experience these teachers have gained is communicated by this Handbook.

This Handbook is not organized in a "read it from cover to cover" basis. After reading this section you will want to browse through the next few pages, which contain brief summaries of each section of the Handbook. You may then want to study Chapter II, which provides a technical overview of the computers in education field. After that you will probably want to read the sections and chapters that best fit your interests and needs. By and large each chapter (and indeed, each section) is independent and self contained. The Handbook was designed this way in order to allow the collaboration of a number of writers and to allow sections to be updated periodically.

SECTION B: SECTION SUMMARIES

This section consists of a brief summary of each section of the Handbook. It serves as an "in depth" table of contents, and as a partial replacement for an index.

I.A: Purpose and Rationale of the Handbook

This Handbook is designed for teachers and school administrators who want to gain "literacy" in the computers in education field. The writing and publication of this Handbook was motivated by a lack of comprehensive up-to-date textbooks designed for use in the computers in education field.

I.B. Section Summaries

This section (the one you are currently reading) consists of brief summaries of each section of the Handbook. (You have now read a summary of a summary--are you in an infinite recursive loop?)

II.A: Technical Overview of COMPUTERS IN EDUCATION

This section gives a brief introduction to a few of the key technical ideas needed to understand the field of computers in education. It defines hardware and software and gives examples of each. It defines and discusses batch processing and time-shared computing.

II.B: Glossary

This section gives a list of somewhat over 100 terms which are frequently used in the computer field. A student completing a "computer literacy" course should be familiar with most of these terms.

II.C: An Overview of Computers in Education

This rather long section gives a good introduction to the field of computers in education. It divides the field into instructional uses of computers and administrative uses of computers. Each category is further subdivided, to provide a picture of all aspects of computers in education, and how these various aspects are related to each other.

II.D: Why Use Computers in Education?

Many schools are not yet started in using computers for administrative or instructional purposes. Others are making inadequate use of the facilities they do have. This article suggests that computers should be used for administrative purposes when they are cost effective. Similarly, computer assisted instruction (CAI) will come into the schools when and where it is cost effective. Teaching about computers (computer literacy and computer programming) and computer aided learning require computing facilities. These instructional uses of computers justify having computers in the schools now.

II.E: Abstracts of Articles

This section contains one page abstracts of about a dozen different recent publications concerned with the general field of computers in education.

III.A: Overview of TEACHING ABOUT COMPUTERS

This section provides a brief overview of Chapter III. The topic Teaching About Computers is divided into three parts. These are Computer Literacy, Computer Programming, and Computer Science.

III.B: Computer Literacy

A person is "computer literate" if he can cope with the computer-related aspects of life in our society. The concept of computer literacy is discussed along with the need for computer literacy. A plan of action for the public schools is given. A computer literacy course at the senior high school level is recommended, and some goals for such a course are listed.

III.C Goals in Teaching Computer Programming

This short section gives a brief history of the teaching of computer programming. It discusses typical goals of a college level introductory computer programming course. It then lists nine possible goals for an introductory computer programming course at the secondary school level.

III.D Uses of CARDIAC

CARDIAC is a Cardboard Illustrative Aid to Computation. It has been widely distributed (free) by Bell Telephone. In the hands of a knowledgeable and skillful teacher it can be a valuable tool for teaching the concepts of computer programming. This article also discusses a number of more advanced applications of CARDIAC to computer instruction.

III.E What Makes A Good Programming Example?

A common complaint of teachers is that they don't have enough good computer program examples for use in class lectures, discussions, and assignments. This section lists criteria for good examples. It presents a detailed format for writing up examples for one's personal library and for distribution. It illustrates this writeup format and discusses its shortcomings. A list of sources of good examples is given in the bibliography.

III.F: Step-wise Refinement, Structured Programming, and BASIC

Step-wise refinement is an essential idea for attacking problems of any significant size or difficulty. Structured programming is a method of incorporating the ideas of step-wise refinement into the programming process. New programming languages need to be developed and made available if structured programming is to become commonplace. A modified version of BASIC is examined, and compared with BASIC. This section is must reading for anyone who wants to teach computer programming!

III.G: Computer Science in the High School Curriculum

The field of Computer Science is defined and various education-related aspects of the field are discussed. It is suggested that a secondary school should offer three types of computer science courses. These courses would be (1) Computer Literacy, (2) Computer Programming and Applications, and (3) Advanced Programming and Computer Structures. A description is given for each course.

III.H. Computer Science at the College and University Level

Computer science is a well established part of the higher education curriculum. A number of publications, such as the ACM issue on "Curriculum 68" give details on appropriate course content. This short section directs the reader to some of these publications.

III.I. Computer Science in High School

This is a detailed discussion of the possible contents and orientation of a year long sequence in computer science at the high school level. Few schools currently offer such a modern and far reaching course.

III.J. Computing on a Shoe String

It is possible for a school or small school district to develop an extensive computer education program with a relatively modest expenditure of funds. An example, along with many excellent suggestions, is provided in this section.

III.K. Abstracts of Articles

It seems as though almost every teacher of computer programming considers himself to be an expert in the general field of teaching about computers. Thus, there are many publications in this area. This section contains one page abstracts of about two dozen current articles.

IV.A: Overview of THE COMPUTER AS AN AID TO LEARNING

Chapter IV discusses the computer as an aid to learning. Computer assisted learning (CAL) can be divided into the situations in which the student does not need to know how to program, and the situations in which the student writes programs. In either case the student makes use of computers while studying some subject matter area. The computer does not present the instruction, as in computer assisted instruction (Chapter V).

IV.B: Simulation

Simulation is one of the more important tools of modern science, business, and industry, and computers are essential to most real-life simulations. This section presents an overview of the field of simulation, with particular emphasis upon computer simulations. It discusses a number of uses of computer simulations in the classroom. Several examples are discussed in detail.

IV.C: Packaged Programs

This section concerns the use of professionally written computer library programs for problem solving. It discusses the educational philosophy of using packaged programs to carry out the computational aspects of problem solving. It gives several examples of uses of such programs. The use of packaged programs is one of the most important parts of uses of computers in education.

IV.D: Information Retrieval

This section defines the area of computerized information storage and retrieval, and discusses its educational implications. Four information retrieval systems, representative of the varying systems found in education, are illustrated and discussed.

IV.E: Computers and Junior High School Mathematics

The author of this section is a talented and innovative teacher of mathematics in a small town junior high school. Some goals of mathematics education are examined in the light of the availability of computers. It is shown that computers can contribute substantially to achieving some of the goals of mathematics education. The existence of computers should cause reexamination of some of these goals.

IV.F: Student-Written Programs for Problem Solving

One of the major current instructional uses of computers is to have students learn some computer programming and then use that knowledge while studying mathematics. At the secondary school level the CAMP series of books and the Colorado Project materials have received considerable attention. The CRICISAM materials are designed to teach a college level calculus-with-computers course. This article provides an overview of the general topic of having students write programs as part of their activity in a regular mathematics course.

IV.G: Abstracts

This section contains abstracts of more than a dozen current articles in the field of the computer as an aid to learning.

V.A: Overview of Computer Assisted Instruction

The computer is an effective interactive instructional delivery system. It must compete with other delivery systems, such as teachers and movies, if it is to gain widespread acceptance for use in this mode. This section presents a brief history and the current status of CAI.

V.B: Computer Assisted Instruction: Some Current Literature

This section presents additional overview material for the topic CAI. It then discusses several reviews of current research findings in the field. These suggest that CAI is an effective educational tool.

V.C: Three Computer Assisted Instruction Systems

This section looks at NEW BASIC, TICCIT, and PLATOIV. These are three of the important and major CAI projects currently going on in the United States. The latter two are receiving extensive federal funding and are undergoing large scale field testing.

V.D: Abstracts of Articles

There is considerable literature in the area of computer assisted instruction. This section contains one page abstracts of more than a dozen current publications.

VI.A: Overview of Computer Managed Instruction

Chapter VI concerns the Computer as a Classroom Management Tool. In computer managed instruction a computer may be used for scoring and recording of homework and tests, diagnostic testing and prescription, monitoring a students status and progress in a course, and individually prescribed instruction. This section gives an overview of CMI. It discusses several examples of current applications of CMI, giving particular attention to projects that are going on in Oregon.

VI.B: Abstracts of Articles

Computer managed instruction is one of the projected "bright spots" for the future of education. Thus the pilot projects have received considerable publicity. This section contains abstracts of about a dozen current publications concerning CMI.

VII.A: Administrative Uses of Computers

This long section provides a comprehensive introduction to the field of administrative uses of computers. It is designed to give administrators insight into the assistance computers might provide them on their jobs. The section contains an extensive bibliography, with particular attention being paid to sources of information in Oregon.

VII.B: Administrative Workshops

A school administrator is often the key individual in deciding whether computers will be available for administrative or instructional use in his school. Workshops to develop computer literacy among school administrators are common. This section outlines the steps to follow in planning such a workshop and describes the content of a recent successful workshop.

VII.C: Abstracts of Articles

It is evident that many school administrators are interested in the field of computers in education. Considerable literature in the computers in education field is written by and/or for administrators. About a dozen current articles are abstracted in this section.

VIII.A: Overview of COMPUTING FACILITIES

The chapter on Computing Facilities is concerned with both computer hardware and computer software. Considerable attention is paid to the problems involved in acquiring appropriate facilities. One section of the chapter discusses various aspects of directing a small computing facility, while another section discusses the components of a quality computing service. The topic of computer networks (especially national networks) is discussed in detail.

VIII.B: Acquiring a Computing Facility

This article is directed at personnel from a secondary school or small college who may be involved in the computer acquisition process. It discusses the steps one should go through in acquiring a computing facility. The key is a careful needs assessment; the use of outside consultants is recommended.

VIII.C: Some Hardware and Software Considerations

If a secondary school wants to acquire a computing facility it may find that it has a large number of logical alternatives to consider. This section discusses some of the major hardware considerations and alternatives. It also discusses the question of software alternatives. Differences in various versions of BASIC are listed and discussed.

VIII.D: The Terminal Facility

This extensive section will be of interest to teachers and school administrators who are involved in the selection and installation of terminal facilities. It discusses the TTY, CRT, and card-reader terminals most apt to be used in an instructional setting. Both technical and practical aspects of terminal selection, installation, and location in the school are considered.

VIII.E: Operating a Small Computing Facility

A small computing facility may consist of a single time-shared terminal, or a one terminal mini computer, or one or two keypunches. This section lists duties and responsibilities of the person who serves as director of such a facility. It recommends "release time" or extra pay for this person.

VIII.F: Components of a Quality Computing Service

The questions of what constitutes good quality computing service, and how one measures it, are very difficult. This section contains a detailed outline of many of the components of a computing service. It serves as a good starting point for those who want to design measures of the quality of a computing service.

VIII.G: Distributed and National Networks

We are in an era of rapid expansion of school district, statewide, and nationwide computer networks. This section defines some of the standard terminology and discusses various sorts of networks. It gives a good comprehensive overview of large network problems and considerations.

VIII.H: Abstracts of Articles

This section contains one page abstracts of about a dozen articles on the general topic of computer facilities.

CHAPTER IX: COMPUTERS IN SPECIAL EDUCATION

This chapter provides an overview of computers in special education. This is a relatively new, and very exciting area of computer application. Many of the applications of computers here will eventually carry over to all of education.

X.A: Abstracts of SURVEYS

Chapter X consists of a collection of abstracts of surveys that are related to the field of computer science education. Each abstract gives a few of the main points in the survey and a reference to the original report.

XI.A: Overview of COMPUTERS IN RESEARCH

Computers are an object of research (developing new hardware, software, applications) and a major tool in research (for example, to process statistical data). This section gives an overview of the role and nature of computers in research. It makes no effort to indicate the nature, extent, or content of what research has actually been done. It does raise a number of possible research questions.

XII.B: A Teacher Training Course

This section gives a detailed outline for an introductory computer science education course. The course is balanced between concepts of computer science education and an introduction to programming in BASIC. It is divided into ten class meetings of about three hours each, to make it suitable for inservice presentation. Such a course would carry 3 quarter hours of college credit.

XII.C: Abstracts of Articles

This section contains abstracts of three publications containing material on teacher training for the computers in education field.

CHAPTER XIII: SOURCES OF ADDITIONAL INFORMATION

This chapter contains sections devoted to additional sources of information. Two of the sections, Places to Visit and Key People, are specifically aimed at Oregon teachers and administrators. The remaining sections are more general, covering Books, Periodicals, Non-print Materials, and Major Computer-Education Projects.

CHAPTER II

COMPUTERS IN EDUCATION

by

David Dempster*

Warren Hall

Ray Korb

David Moursund*

Stew Weimer

*David Dempster, Computing and Mathematics specialist, Lincoln County, Oregon, served as group leader for this chapter, and David Moursund served as editor.

SECTION A: TECHNICAL OVERVIEW

David Moursund

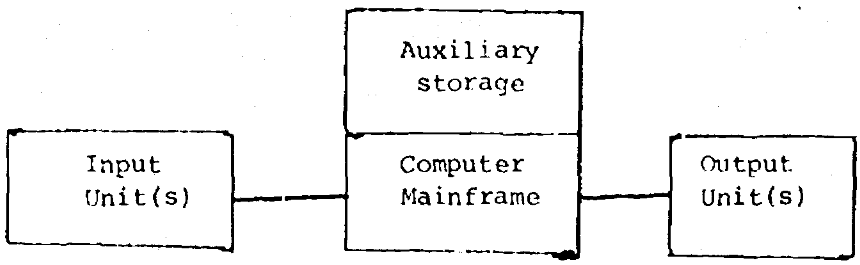
Introduction

It is hoped that the title of this section will not keep the reader from getting through this first sentence! And, once started, we hope he will continue reading... Computer science is a field in which one studies computers and their uses. Some aspects of the field are highly technical; other aspects involve insight into application fields such as business, engineering, the physical sciences, or the social sciences. This section of the handbook will not make you into a computer scientist. However, it will introduce you to a few of the major ideas and problems of computer science. Note that Section B of this chapter contains an extensive glossary of terms that tend to be used in discussing the field of computers in education, while Section C gives an overview of computers in education.

A computer is a machine which consists of hardware (physical machinery) and software (computer programs). Hardware and software are two of the key ideas in the computer science field, and we shall discuss each in some detail. Then we shall discuss two classifications of computer systems--time shared systems and batch processing systems. Finally, we shall summarize the key ideas discussed in this section.

Computer Hardware

An electronic digital computer system is a machine designed for the input, storage, manipulation, and output of information. It can carry on this task automatically, under control of a set of directions (called a computer program) stored in the computer's memory. By "information" we mean anything that can be expressed using the letters, digits and punctuation marks of a natural language (i.e., English) character set. In this section we will discuss the various types of physical machinery (hardware) found in a computer system. A block diagram of the computer system components to be discussed is given below. These components may be classified as peripheral equipment (input units, output units, and auxiliary storage units) and mainframe (memory, control unit, arithmetic and logic unit).



Input Unit The ubiquitous punched card, for storing information, has been with us since the United States census of 1890. It was at that time that Herman Hollerith developed the ideas and equipment needed to use punched cards for information processing. Information is "punched" into cards using a keypunch, which is a device much like an electric typewriter. Punched cards are read into a computer using a card reader, which is one of the standard computer input devices.

Another important computer input device is a keyboard terminal. One can think of this as an electric typewriter connected directly to the computer mainframe. Such a keyboard terminal is also a computer output unit.

Output Unit Almost all of us have received computer-prepared bills, either in the form of punched cards or printed on appropriate forms. Anything that can be stored in a computer's memory can be outputted onto punch cards, or can be printed out by a line printer. Note that a high speed computer driven line printer is several hundred times as fast as a good typist!

Auxiliary Storage Almost everybody has seen pictures of magnetic tape units used in computer systems. These are somewhat similar to those used in home tape recorders, but are of greater quality and precision. Another type of auxiliary storage is a disk or disk pack. A disk is a circular plate (think of a phonograph record) whose surfaces are covered with magnetizable material. Several of these may be stacked together, separated by air spaces, to form a disk pack. On disk packs, and on magnetic tape, information is stored as magnetized spots using an appropriate code. Typically the auxiliary storage devices in a computer system can contain much more information than the main computer memory, and at a much lower cost. Auxiliary storage devices are generally much slower than main computer memory in terms of time needed to store or retrieve a piece of information.

Mainframe The computer mainframe consists of the main memory, control circuitry, and the arithmetic and logic unit. The size of the main memory is one key factor in determining the cost of a computer system. A small computer mainframe will have a memory which can store about 4000 characters (i.e., letters, digits) of information. A very large computer system may have a main memory that can store several million characters of information.

The control unit of a computer is the circuitry which allows the computer to follow a step by step set of directions automatically. That is, a computer program is a step by step set of directions for carrying out a particular information manipulation task. When this step by step set of directions is coded in appropriate form and placed in a computer's main memory then the control unit can cause these directions to be executed (followed).

The arithmetic-logic unit carries out the arithmetic and logic operations needed to process information. It operates under the control of the control unit (which in turn follows the directions given in a computer program). One measure of a computer's "size" is the speed of its arithmetic and logic unit. A modern "slow" computer may be able to carry out 50,000 to 100,000 operations per second. A "fast" computer may be able to carry out 10 million to 100 million operations per second.

To summarize, a computer system contains hardware which may be classified as peripheral equipment (input/output units, auxiliary storage) and mainframe. The overall system is designed for the input, storage, manipulation, and output of information. We can think of information as anything that can be typed or coded using a standard typewriter. Two short definitions of computers are automated character manipulator, and automated information processor.

Computer Software

We have just discussed the fact that computer hardware is the physical machinery needed to input, store, manipulate and output information. Computer software is the computer programs that guide or direct the hardware in carrying out such tasks or assist in preparing computer programs. We will discuss three types of software: systems software, applications software, and user-written programs. There are two key concepts to keep in mind. (1) A computer system is useless without appropriate software. (2) Software costs money (it takes considerable time and expertise to develop).

Systems Software A computer is built to "understand" a particular instruction set called a machine language. Each model of computer has its own machine language, and this may consist of 50 to 300 or so different instructions. Typically these will include the operations of addition, subtraction, multiplication, division, various logical operations, operations needed to run input and output equipment, etc.

It is possible (but very slow and frustrating) to write programs directly in a machine's machine language. This is rarely done. Rather one writes programs in higher level languages and appropriate computer programs (systems software) translate these programs into machine language. At the most elemental level one writes programs in assembly language. Here one can use mnemonics such as ADD (for add) and SUB (for subtract), and variable names such as HOURS and RATE to express and keep track of what one wants to have the machine do. A program written in assembler language is translated into machine language by a computer program (software) called an assembler.

At a higher level there are a number of procedure-oriented or problem-oriented languages such as BASIC, COBOL, and FORTRAN. These languages tend to be fairly easy to learn, and are designed to aid the user in expressing the steps needed to solve problems in a wide variety of areas. BASIC is essentially a student oriented language. A sample program, to compute the area and perimeter of a rectangle, is given below

```
10  LET L = 28.95
20  LET W = 16.37
30  LET A = L*W
40  LET P = 2*L+2*W
50  PRINT "AREA IS", A
60  PRINT "PERIMETER IS", P
70  END
```

COBOL is designed for business oriented problems, while FORTRAN is a general purpose science-oriented language.

For each higher level language, and each model of computer, it is possible to write a piece of software (called a compiler or translator) which can translate programs written in the higher level language into the machine's machine language. The software itself, for just one translator, may require several man-years of effort to develop.

The key concept in systems software is that there are literally hundreds of computer languages available. When a new model computer is first manufactured, software for it must be written. Many computer systems have only an assembler and no higher level languages. Others may have an assembler and BASIC, but not COBOL. When one acquires a computer system he must pay equal attention to the hardware and the software facilities he is acquiring.

Applications Software It is possible for a person to use a computer program written by another person. Thus it is feasible--and indeed, highly desirable--to construct a library of programs designed to solve a wide variety of problems. These programs are examples of applications software. A good library will include dozens of different statistical analysis programs, programs to solve ordinary and differential equations, programs to do mathematical curve fitting, etc.

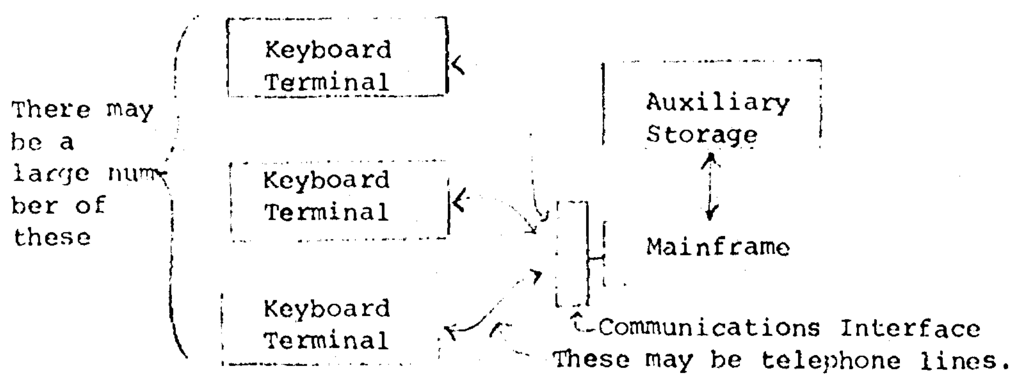
A computer assisted instruction (CAI) package would be an example of applications software. A good CAI package will contain very extensive programs designed for the presentation of instructional material, the recording and analysis of student responses, and so on. It will also contain programs designed to help people create CAI lessons.

User-Written Software Most people, given appropriate instruction, can learn to write computer programs. For the most part a student will write a program to solve a specific problem, use the program and then discard it. This is also fairly common in fields like science and engineering, where thr practitioner may make frequent use of computers. Such programs, which tend to have a rather short life span and tend to be used only by the program writer, are referred to as user-written software.

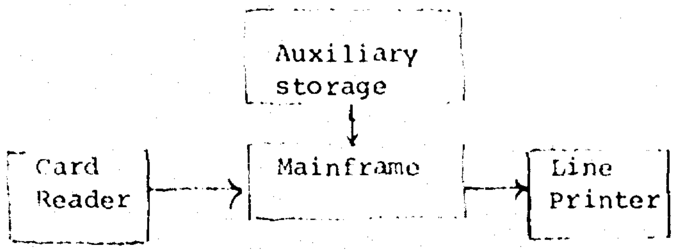
To summarize, it is important to understand that a computer cannot solve a problem unless it is directed by an appropriate computer program. Some computer programs (systems software) are designed to aid people in writing computer programs and/or to aid people in running a computer system. Other programs are designed to solve particular types of problems, and are collected together in an applications library. When one acquires a computer system he acquires both hardware and software. The quantity and quality of the systems and applications software will often be the key factors in deciding what system to acquire.

Time-Shared and Batch Processing Computing

There are two general classifications of computer systems, based upon the type of input/output equipment they support. Block diagrams showing the major features of each system are given below:



A Time-Shared Computer System



A Batch Processing Computer System

A time-shared computer system's hardware and software are designed so that a number of people can make simultaneous use of the system. Thus several users may be typing in programs, several others may be waiting while their programs are processed and still others could be receiving output on their terminals. The great speed of a modern computer system makes this possible. The computer divides its attention among the various terminal users, switching rapidly from one user to the next. To illustrate, suppose we had a computer that could perform 1 million arithmetic and/or logic operations per second, and that twenty users were active on the system. In one second the machine could devote 50,000 operations worth of compute-capacity to each user. Many of the users, however, will not need nearly this much computing done in a typical second. They will be typing in programs, or data, at a slow rate, and will require very little of the computer's attention. This means that the users whose programs are being executed can receive a greater fraction of the machine's capacity, perhaps several hundred thousand operations per second.

In a batch processing computer system one person uses the computer at a time. His program and/or data to be processed are placed in the card reader so they can be read into the computer's memory. Typically the data consists of a batch of similar transactions or problems to be processed. Thus the program might be designated to prepare a bill, and one computer run might be used to process a batch of several hundred bills.

The two "models" discussed above are over simplified, and there are many variations on these two themes. For example, a modern minicomputer (say one costing $5,000) is apt to have a single keyboard terminal as its input/output device, and its only auxiliary storage may be a paper tape punch/reader located on the keyboard terminal. Such a system would be classified as a batch processing system. It could be used to run an interactive language such as BASIC, which is typically considered to be a time-shared language.

Another variation would be to add a card reader and line printer to a time-shared system. The resulting system then functions simultaneously as a batch processing and as a time-shared computer system. Note that it takes both appropriate hardware and appropriate software to be able to do this.

A key concept in any case is that a computer system can be used for all kinds of purposes. Either a batch processing system or a time-shared system can be designed to satisfy the administrative and instructional needs of a school. However, some computer systems are designed mainly for instructional purposes while others are designed mainly for business and/or administrative purposes. A computer system designed primarily for one type of application is apt to be inefficient or even completely unsuited to some other type of application. A small time-shared system for instance, may be specifically designed for instructional use, providing time-shared BASIC at a low cost per terminal hour. Such a system would probably prove unsuitable for many school administration uses. Similarly, a small batch processing system may be quite suited to a school's administrative computing needs, but be quite unsuited to a general instructional environment. Colleges and universities tend to try to acquire computer systems that are balanced between the instructional, research, and administrative needs of the school. A single sufficiently large computer system (with appropriate hardware and software) can provide high quality service to all three types of educational users.

Conclusion

This section of the Handbook gives a brief introduction to the ideas of computer hardware and software, and of batch processing and time-shared computing systems. Key ideas include:

1. A computer is an information processing machine.
2. A computer system consists of hardware and software.
3. Computer systems are generally classified as batch processing or time-sharing.
4. A computer system may be designed to fill a single educational need such as instruction or administration, or may be designed to fill all of a school's educational needs.

Many of the ideas touched upon in this section are expanded in subsequent sections. The glossary given in the next section defines many of the terms used in this Handbook.

SECTION B: GLOSSARY

Each field of human intellectual endeavor has its own vocabulary, and the field of computer science is no exception. A number of computer literacy and computer appreciation text books were examined in order to select terms that are considered essential to a minimal, non-technical, understanding of the computing field. A student completing an introductory computer literacy course should have a working knowledge of most of these terms.

Access time. Time required to read out or write in data from a a data storage system. This is an important characteristic of a storage system. Generally speaking the shorter the access time, the more costly the storage system.

ALGOL. A computer language designed mainly for programming scientific applications. This is one of the more modern, and widely used compiler languages (i.e., procedure-level languages).

Algorithm. A finite, step-by-step set of directions designed to solve a particular type of problem.

Alphanumeric character. The letters, numbers, punctuation marks, and other special symbols that a given computer's input/output circuitry is constructed to handle. Alphanumeric characters are also known as Hollerith characters. (Herman Hollerith pioneered the use of punched cards for data processing).

Analog computer. A computer in which analog representations of data are mainly used. For example, voltages or currents might be used to represent the variables in a differential equation to be solved by an analog computer.

Application program. The software for a computer system may be classified as Applications Programs and Systems Programs. An Application Program is designed to solve a certain type or class of problems. For example, one might have an Applications Program designed to solve a certain type of equation, or to perform a specific statistical computation.

Arithmetic unit. The portion of the hardware of a computer in which arithmetic and logical operations are performed. One can think of it as a superspeed electronic calculating device.

Artificial intelligence. The branch of computer science concerned with the study of the possibility of, methods of, and implications of developing computer systems which can perform intelligent-like tasks such as interacting in a natural language, game playing, question answering, theorem proving, etc.

Assembler. A computer program that takes instructions written in assembly language and converts the instructions into the language required for operation. It is an example of systems software.

Assembly language. A computer language intermediate between machine language and compiler languages. It allows machine-language instructions to be written in simplified form using mnemonics and other standardized abbreviations. For example, mnemonics like ADD and SUB are used in place of their numeric codes.

Auxiliary storage. A peripheral storage device that can store information in a form acceptable to the computer, such as magnetic tape, magnetic disks, and magnetic drums.

BASIC. Beginner's All-Purpose Symbolic Instruction Code. A procedure-level computer language that is well suited for time-sharing. It is one of the easiest computer programming languages to learn, and is designed to be used in a time-shared computer environment.

Batch processing. A systems approach to processing where a number of similar input items are grouped for processing during the same machine run. It is generally associated with a single person using a computer at one time (for several seconds or a longer time period) as contrasted with time-shared computing in which many users appear to be making simultaneous use of a machine.

Binary. Compounded or consisting of two things or parts. Binary arithmetic is a system of arithmetic making use of the two symbols 0 and 1. In binary arithmetic the arithmetic operations are performed using the number base 2.

Binary digit. See bit.

Bit. A coined word from binary digit; this is one of the whole numbers, 0 or 1, in a single position, in the binary scale of notation.

Bug. An error in a computer program.

Byte. Depending upon the make and model of computer, a single character in a computer's memory will require a specific number of bits of storage. This unit of storage is called a byte. Common byte lengths are 6 to 8 bits.

CAI. Computer Assisted Instruction. A method of using a computer system as a means of presenting individualized instruction materials. In CAI the computer is used as an instruction delivery device.

CAL. Computer Augmented Learning. A method of using a computer system to augment, or supplement, a more conventional instructional system. A typical example would be using canned programs to aid in the problem solving process in a course of instruction.

Card punch. A computer output device that punches holes in cards.

Card reader. A device that inputs the information on a keypunched or mark sense card into the computer.

Cathode ray tube. A vacuum tube such as a picture tube in a television. When used as a computer terminal, it displays output on a TV-like screen. A keyboard terminal containing a cathode ray tube is often called a CRT terminal.

Central processing unit. The group of components of a computer system that contains the logical, arithmetic, and control circuits for the basic system.

Character. A letter, digit, punctuation mark, or other sign used in the representation of information. Computers are designed for the input, storage, manipulation, and output of characters.

COBOL. A computer language designed mainly for programming commercial applications. This is one of the most widely used compiler languages (i.e., procedure-level languages).

CMI. Computer managed instruction. An application of computers to instruction in which the computer is used as a record keeper, manager and/or prescriber of instruction.

Compiler. A computer program (i.e., software) that translates a program written in a procedure-oriented language such as BASIC, COBOL, or FORTRAN into a machine language or an assembler language.

Compiler language. A language such as BASIC, COBOL, FORTRAN, etc. designed to assist the programmer in writing procedures to solve problems.

Computer. A device capable of accepting information, applying prescribed processes to the information, and supplying the results of these processes. It usually consists of input and output devices, storage, and communications units, and a central processing unit.

Console. That part of a computer used for communication between the operator or maintenance engineer and the computer. It is rougly comparable (but generally more complicated) to the keyboard on an electronic calculator).

Control unit. A unit or portion of the hardware of an electronic digital computer that is designed to direct a sequence of operations, interpret coded instructions, and initiate proper commands to the computer circuits. It is part of the central processing unit.

Core storage. The main or integral memory of a computer. The word comes from the very small donut shaped iron cores which, at one time, were the most widely used form of memory. Nowadays other, solid state, devices are often used as main memory--but may still be mistakenly called core memory.

Cybernetics. The study of control and communication in man and the machine. The name of Norbert Wiener is most often associated the development of this field.

Data. Detailed information such as facts, numbers, or quantities used to solve some problem or reach some conclusion. Information to be processed by a computer program.

Data bank. A comprehensive collection of libraries of data. For example, one line of an invoice may form an item, a complete invoice may form a record, a complete set of such records may form a file, the collection of inventory control files may form a library, and the libraries used by an organization are known as its data bank. The term is often associated with a collection of information about people, and the possible misuse of computers.

Data preparation. The process of organizing information and storing it in a form that can be input to the computer.

Data processing. The execution of a systematic sequence of operations performed upon data. (Synonymous with "information processing").

Debug. To remove all malfunctions or mistakes from a device, or, more usually, from a program.

Digital computer. (1) A computer in which discrete representation of data is mainly used. (2) A computer that operates on discrete data by performing arithmetic and logic processes on these data. (Contrast with "analog computer").

Digitize. To render a continuous or analogue representation of a variable into a discrete or digital form.

Direct access. The ability to read or write information at any location within a storage device in a nearly constant amount of time. A computer's main memory is usually a direct access storage device. A magnetic disk is another example of a direct access storage device.

Disk storage. A storage device that uses magnetic recording on flat rotating disks. It is a direct access storage device.

Drum. A rapidly rotating cylinder, the surface of which is coated with a magnetic material on which information is stored in the form of small magnetized spots. It is an example of a direct access storage device.

Electronic digital computer. A computer in which data is represented in digital form.

Execution time. That period of time while a program is in execution. Also the total time required to execute an entire program.

Feedback. A means of automatic control in which the actual state of a process is measured and used to obtain a quantity that modifies the input in order to initiate the activity of the control system.

File. A collection of related records treated as a unit. For example, one line of an invoice may form an item, a complete invoice may form a record, a complete set of such records may form a file, the collection of inventory control files may form a library, and the libraries used by an organization are known as its data bank.

First generation computer. Computers whose circuitry depended heavily upon vacuum tubes. The vacuum tube era ended about 1958, when transistorized computers began to be produced.

Flowchart. A chart consisting of various flow diagram symbols such as arrows, rectangular boxes, circles, and other symbols used to graphically represent a procedure or pattern of computation to solve a particular problem.

FORTRAN. A computer language designed mainly for programming scientific applications. The initial FORTRAN was developed by IBM during 1953-57. It is still one of the most widely used compiler languages (i.e., procedure-level languages).

Hardware. The electrical, electronic, magnetic and mechanical devices or components of a computer. Typically one speaks of a computer system as consisting of hardware and software, where hardware includes the I/O devices, the central processing unit, main memory, and the peripheral devices such as auxiliary storage devices.

Heuristic. Trial-and-error method of tackling a problem, as opposed to the algorithmic approach. Many of the problems used in the field of artificial intelligence are heuristic programs.

Hollerith character. See alphanumeric character. Herman Hollerith pioneered the use of punched cards in data processing (and in particular, in the processing of the 1890 U.S. census data).

Information. In reference to computers, information is the same as data. One can think of it as anything that can be coded or represented using the character set one has available.

Information retrieval. A branch of computer sciences relating to the techniques for storing and searching large or specific quantities of information.

Input. Information which a computer system's elements receive from outside the computer system.

I/O. Input and/or output.

I/O device. A unit that accepts new data, sends it into the computer for processing (input), receives the results from the computer, and converts them into a usable form. (output)

Instruction. A coded program step in a programming language, such as in a machine, assembler, or compiler language.

Interface. The point of contact between different systems or parts of the same system. One of the most important aspects of the field of computer science is the man-machine interface problem.

K. The letter K is used to represent the number 2^{10} which is 1024. The size of a computer's memory is often stated in terms of a number of K of words or bytes. Thus a small computer memory might be 1K or 8K words, while a large computer memory might be 96K or 128K words, or more.

Keyboard terminal. An electric-typewriter-like computer I/O device.

Keypunch. A keyboard device for punching holes in cards.

Line printer. A printer where an entire line of characters is composed and determined within the device prior to printing. Thus a whole line is printed nearly simultaneously; speeds of 300-1800 lines per minute are common, as are line lengths of 80 to 132 characters.

Loop. The repetitious execution of a series of instructions caused by having the last instruction in the series return the machine to the first instruction in the same series.

Machine language. The language, or instruction set that a computer is constructed to "understand" or be able to perform.

Magnetic core. A data storage device based on the use of a highly magnetic, low-loss material, capable of assuming two or more discrete states of magnetization. See core storage.

Magnetic disk. A flat circular plate with a magnetic surface on which data can be stored by selective magnetization of portions of the surface. It is considered to be a random access storage device.

Magnetic drum. A data storage device using magnetized spots on a magnetic rotating drum; permits quasi-random medium-speed access to any part of its surface.

Magnetic ink character recognition. (MICR) The machine recognition of characters printed with magnetic ink. (Contrast with OCR). A standard application of MICR is on checks.

Magnetic tape. A device for storing digital or analogue data in the form of magnetized areas on a tape of plastic coated with magnetic iron oxide. Generally considered to be a sequential access storage device.

Mean time to failure. The average length of time for which the system, or a component of the system, works without fault.

Memory. A computer's storage device.

Memory location. See storage location.

Microsecond. One-millionth of a second. A small or medium scale modern computer can add two numbers in a few microseconds.

Millisecond. One-thousandth of a second. The speed of auxiliary storage devices (their access time) is usually measured in milliseconds.

Monitor. A program to supervise the proper sequencing of programming tasks by the computer. It is an example of computer software, and is often called an operating system.

Nanosecond. One thousand-millionth of a second (i.e., a billionth of a second). A very fast, modern computer can add two numbers in less than 100 nanoseconds.

Numerical analysis. The branch of mathematics in which one studies the numerical procedures needed to solve mathematical and engineering type problems on a computer.

Numerical control. A means of controlling machine tools through servo-mechanisms and control circuitry, so that the motions of the tool will respond to digital coded instructions on tape.

OCR. Optical character recognition. Machine identification of printed characters through use of light-sensitive devices. (Contrast with MICR).

On line. An on-line system may be defined as one in which the input data enter the computer directly from their point of origin and/or output data are transmitted directly to where they are used. The intermediate stages such as punching data onto card or paper tape, writing magnetic tape, or off-line printing, are largely avoided.

Optical mark reader. An input device that reads graphite marks on cards or pages.

Output. Information which a computer systems transmits via an output device such as a terminal, line printer, card punch.

Peripheral equipment. Ancillary devices under the control of the central processor, eg. magnetic tape units, printers, or card readers.

PL/I. A computer language designed for programming both scientific and commercial applications. One of the more modern compiler languages.

Process control. Use of a computer system to control a process, such as the operation of part of an oil refinery.

Program. A set of coded instructions to direct a computer to perform a desired set of operations. Generally a program is designed to solve a particular type of problem or carry out a specified task.

[illegible]rammer. A person who prepares programs. That is, a person who prepares problem solving procedures and flow charts and who may also write and de-bug programs.

Program library. A library of computer programs; each program in the library is designed to solve a certain type (class) of problem.

Punch card. Thin cards on which digits are represented by holes in selected locations for storing data.

Punched tape. A paper or plastic tape in which holes are punched to serve as a digital storage device.

Punched card. A card punched with a pattern of holes to represent data. (A punch card is the same card before it is punched. In both cases the card is frequently called an IBM card and occasionally is called a Hollerith card.)

Random access. Access to data storage in which the position from which information is to be obtained is not dependent on the location of the previous information, e.g. as on magnetic drums, disks, or cores. The time required to access a piece of information is nearly constant (nearly independent of the location of the information).

Real time. A real-time computer system may be defined as one that controls an environment by receiving data, processing them and returning the results sufficiently quickly to affect the functioning of the environment at that time. Generally this means that the computer responds to the situation in a quite short period of time.

Record. A group of related items (i.e., a group of related pieces of information). Records are, in turn, grouped together to form files.

Remote terminal. A device for communicating with computers from sites which are physically separated from the computer, and usually distant enough so that communications facilities such as telephone lines are used rather than direct cables.

Response time. This is the time the system takes to react to a given input. If a message is keyed into a terminal by an operator and the reply from the computer, when it comes, is typed at the same terminal, response times may be defined as the time interval between the operator pressing the last key and the terminal typing the first letter of the reply. For different types of terminal, response time may be defined similarly. It is the interval between an event and the system's response to the event.

Second generation computers. A computer belonging to the second era of technological development of computers when the transistor replaced the vacuum tube. These were prominent from 1959 to 1964, and were displaced by computers using integrated circuitry and large scale integrated circuitry.

Sequential access. A process which consists of reading or writing data serially, and by extension, a data recording medium that must be read serially, as a magnetic tape.

Simulation. The representation of physical systems and phenomena by computers, models, or other equipment. When information enters the computer to represent factors of a process, the computer produces information that represents the results of the process.

Software. The computer programs part of a computer system. Generally divided into systems programs, and applications programs. Although the term most often applies to the "professionally written" general purpose programs in a computer system, it sometimes is used to denote any computer program.

Source program. A program that must be translated into machine language before use. Thus, a program written in a procedure oriented language such as BASIC or COBOL would be called a source program.

Storage, auxiliary. A peripheral storage device that can store information in a form acceptable to the computer, such as magnetic tape and magnetic disks.

Stored program computer. A computer than can store a program in its memory--i.e., the typical modern electronic digital computer.

Symbol. In computing, a character. That is, an element of the computer's character set.

System. An assembly of components united by some form of regulated interaction to form an organized whole. Also a collation of operations and procedures, men and machines by which an industrial or business activity is carried on. A computer is a system consisting of hardware and software.

Systems program. The software for a computer system may be classed as Applications Programs and Systems Programs. The Systems Programs include assemblers, compilers, and monitors or operating systems.

Teleprocessing. A form of information handling in which a data processing system utilizes telecommunication facilities.

Terminals. The means by which data are entered into the system and by which the decisions of the system are communicated to the environment it affects. A wide variety of terminal devices have been built, including teleprinters, special keyboards, light displays, cathode tubes, thermocouples, pressure gauges and other instrumentation, radar units, telephones, and so on.

Third generation computers. Computers which use large scale integrated circuitry and miniaturization of components to replace transistors, reduce costs, work faster, and increase reliability. The third generation of computers began about 1964.

Time-sharing. A method of operation in which a computer facility is shared by several users concurrently.

TTY. Abbreviation for a Teletypewriter keyboard terminal.

Turnaround time. The amount of time that is required for a computational task to get from the programmer to the computer, onto the machine for execution, and back to the programmer in the form of the desired results.

Word. A set of characters or bits which is handled by the computer circuits as a unit.

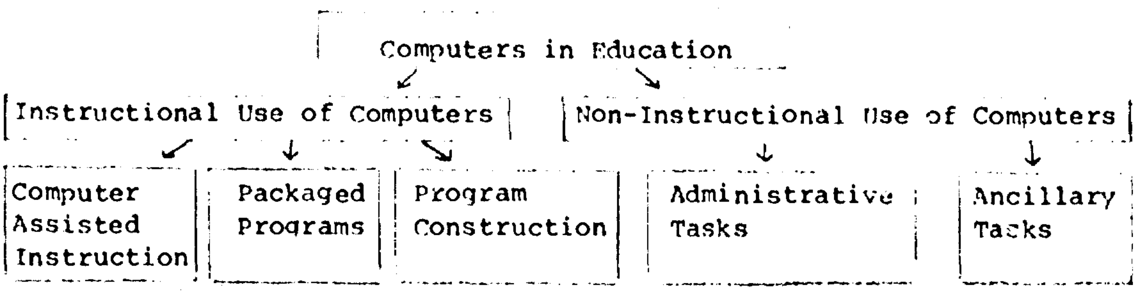
Word length. The number of bits or characters in a word. On an IBM 360 or 370 computer the word length is 32 bits. On a Digital Equipment Corporation PDP-10 the word length is 36 bits. Many minicomputers have word lengths of 16 bits.

Section C: AN OVERVIEW OF COMPUTERS IN EDUCATION - David Dempster

Introduction: Each year more and more schools are investing considerable resources in the use of computers for various educational tasks. This article presents a general overview of the many ways computers are currently used in educational settings and what one can expect to see in the near future.

Much of the early work in developing computers was carried out by universities and thus much of the early use of computers was in an educational environment. Usage gradually spread from high level research to the undergraduate curriculum. By the early 1960's the use of computers had extended down to include some high schools. This downward trend is still continuing and is now starting to make an impact on elementary schools. Although the educational use of computers lags several years behind the use in other fields (business, industry, government) in terms of breadth and depth, it will have a profound impact on education during the next decade.

The field of educational computing can be separated into two general areas--instructional and non-instructional. The following diagram shows this division and the subdivisions of each major category.



The major characteristic used (for the purposes of this article) to determine in which category a given educational task is placed is that of who interacts directly with the computer. If a student submits programs and/or data to the computer and/or receives output from the computer, the the use is considered instructional. Otherwise, it is non-instructional.

Instructional Use of Computers. The instructional uses of computers is slowly becoming a reality in education. Computer facilities for students are already quite common in higher education and are becoming more common in secondary schools. In Oregon, for example, approximately 3/4 of the high school students attend high schools that have (at least some) instructional computing facilities. Typically this is one or two time-shared terminals in a school of 1000 to 2000 students. At this time facilities in elementary schools are scarce. Because the applications of computers to instructional problems are almost unlimited, however, the use of computers will continue to grow.

A good way to look at instructional uses of computers is to classify them on the basis of student knowledge required to use the computer. Very little knowledge is required for certain types of computer assisted instruction (CAI) or testing students with computers. Even first graders have been taught to interact with a computer. A good example of this is provided by the CAI materials of Pat Suppes of Stanford University.

At a somewhat more sophisticated level we have the students using packaged ("canned") programs (programs developed by someone else) to solve problems and study some situation or process. In these cases the program is supplied by "experts" and the student runs and interacts with it.

At a higher level of student knowledge we have the student actually creating and running his own programs. Most of the current instructional use of computers at the secondary school level is of this type.

Computer Assisted Instruction (CAI): One can think of CAI as a delivery system. (See Chapter V of this Handbook for a more detailed discussion of CAI.) It competes with books, television, movies, filmstrips, slides, video and audio tapes, teacher lectures, etc. The typical CAI setup involves a computer terminal hooked to a computer (often via a telephone line). The terminal may include both audio and/or video capabilities. The key concept is one of interaction between the student and the CAI system. This allows both good and rapid feedback to students, and the individualization of the learning activities. The system can keep accurate, detailed records of many students at one time.

CAI in the past has been examined in a few major projects, such as those at Stanford, University of Texas, and University of Illinois. In addition to such major projects there have been many small CAI projects (usually just drill and practice) that individual teachers or schools have developed. This grass-roots develoment stems from the fact that small, simple CAI systems are relatively easy to construct. Most of this activity is under the heading of research and development. Some CAI systems have been fairly well tested, and have proven to be competitive with other systems as far as student learning is concerned. By and large CAI has been (is) more expensive than traditional educational delivery systems.

Continued research and development in the field of Computer Assisted Instruction, along with decreases in the cost of computer hardware, are making CAI more cost/effective. Already the estimates of the costs of CAI in a large modern system are on the order of 50¢ per hour of student usage. Although this is expensive when compared to just a textbook and a teacher (less than 30¢ per hour at the pre-college public school level) one must keep in mind the anticipated advantages of a sound individualized program with rapid feedback and evaluation. As the costs of CAI decrease and the advantages of the approach are proven CAI may eventually replace certain traditional delivery systems. It is worth noting that the cost per hour of a modern CAI system is considerably below the cost of many special education programs for slow learners, handicapped students, etc. CAI is already cost effective in more expensive higher education programs such as medical schools.

Teachers and administrators need to keep an open mind concerning CAI. They should compare and contrast it against other delivery systems when it becomes available. Since CAI may replace and/or supplement some classroom activities, teachers need to learn how to make effective use of this new educational tool.

Packaged Programs: A packaged program is a computer program which has been prepared by some "expert" to perform some specific task. There are two general classes of packaged programs. They are 1) those which perform a task for the user and 2) simulations. (See Chapter IV of this Handbook for a more detailed discussion of these topics.)

There are a great many computer programs developed by "experts" to perform tasks that "non-experts" want done but lack the ability or time to do. For example, there are many statistical packages which perform statistical procedures on a set of data, such as test scores. All the user needs to do is to supply the data which needs to be processed. Another example would be a package which would solve certain types of mathematical equations. Both of these examples free the student from the "grind" work and allow him to use the results of procedures which would otherwise be unavailable to him.

Computer simulations are a type of packaged program which allow students to experience situations and problems not normally available in the schools. A simulation is a computer program which usually involves several parameters which the student can manipulate for the purpose of seeing how the changes affect the outcome of the program. For example, a computer program can simulate the process of pollution of a river. The parameters might be the types of sewage, the type of treatment the sewage receives before it is dumped into the river, the temperature of the water, and the amount dumped into the river each day. The student has control of these parameters and may experiment to see how they interact with one another and affect the result. (The Huntington II project has produced such a simulation; it is called POLUT, and is written in BASIC.)

Computer simulations range in size and complexity from rather small, simple ones (such as simulating the height of a bouncing ball) to large, complex ones (such as one recently developed to simulate the buying habits of the American consumers). The Huntington II project (Polytechnic Institute of Brooklyn) is developing a sequence of excellent simulations written in BASIC, and designed for instructional use. These are being distributed commercially by Digital Equipment Corporation.

At present there is little research about the use of packaged programs in the curriculum. Although there are many articles expounding the advantages of packaged programs, especially simulations, there is little valid research evidence to be found in the literature. With respect to packaged programs there are a number of things educators need to know. 1) Do they lead to a more rapid assimilation of "facts"? 2) Do they deepen a student's understanding of a subject or process? 3) Do they assist students in solving problems that were previously too difficult for them to solve? 4) Do they affect a student's attitude about a subject and/or school? There are a host of other questions that need to be answered. Part of the difficulty in doing the research is to separate clearly the cause-effect relationship. Also, the comput . `` so new to the curriculum that the novelty of using it may obscure hat s really happening. Another problem is that many of the advantages cla ned for packaged programs lay in the area (problem solving, attitudes and un rstanding) in which changes are extremely difficult to assess with present evaluative instruments.

As computers become more readily available to teachers and students, the use of packaged programs will expand rapidly. As the use of packaged programs expands it will be necessary for teachers to constantly re-evaluate their current classroom activities to determine if the use of packaged programs can aid their overall instructional program. In addition, the goals of instruction may need to be evaluated in light of the greater range of concepts, generalizations and procedures packaged programs make available.

Program Construction: Program construction (programming) requires the highest level of student knowledge about computers. (See Chapter III of this Handbook for a more detailed discussion of ideas related to teaching computer programming.) At this level the student actually has control over what the computer does and how it goes about doing it. Computer programming is much more than just coding instructions in some computer language that is acceptable to a given computer. The following steps give a general idea as to what one needs to do in order to use a computer to solve a specific problem.

1. Clearly define the problem.
2. Analyze the problem and determine the most feasible method of solution, keeping in mind what a computer is capable of doing.
3. Design a solution, usually in very general terms so all similar problems can also be solved with the same program or with slight modifications.
4. Code the solution into some programming language.
5. Enter the program into the computer along with the appropriate data.
6. Debug the program for logical and syntax errors.
7. Run the program to solve the problem.
8. Analyze the results in light of the original problem.

These steps show all the traditional aspects of problem solving with a new dimension of using the computer.

Computer programming is often introduced into the curriculum for one of two reasons. First, it may be the subject of study in its own right. This may be a one term, semester, or year course taught in the science or mathematics department. The major goals are to be able to program the computer in some language and learn about how computers function. The second type of introduction to computer programming is for the purpose of giving the students a powerful tool to use in the study of some subject such as science, mathematics, or business. In this aspect program construction is tied directly to some existing part of the curriculum to enhance the learning of that subject. The "Colorado Project", a 2nd year algebra and trig course, is a good example of this.

To be able to offer computer programming in a school it is necessary to have 1) a teacher who can program a computer in a language which is suitable for students and 2) access to some computer facilities. It is possible to "write" programs by sorting a stack of prepunched cards containing a variety of commands, and then submit these programs via US mail to a computer. This would be an example of computing using ultra-minimal facilities. Better facilities would include a keypunch, better access to a computer, or time-shared terminals. The most common instructional computer access now-a-days in the public school is a time-shared terminal.

As in the case of prepackaged programs there is little research backing up the claims of the enthusiasts for teaching computer programming. Two questions which are especially in need of more research are: "Are students who have learned to program the computer better problem solvers when the computer is not available than those who have not learned to program?" and "Do students learn a subject better because they have learned to construct computer programs for the study of that subject?" Note that we do not dispute the fact that students can be taught to program a computer, and that this may be a useful skill in its own right.

Teachers need to examine their already crowded curriculum in light of the anticipated advantages of problem construction to determine if programming is worth implementing. If they decide it is worth implementing then they need to determine how and where such implementation should take place.

Instructional Facilities: To some extent the type of facilities available in the school will determine the type of instructional uses to which the computer can be put. It is necessary to understand the two general types of facilities and what can be done on each before thes relationship is clear. The two general types of facilities are "batch" and "on-line".

Batch systems usually consist of a computer with a card reader for input and a line printer for output. Programs are submitted to an operator who runs the programs through the computer at a convenient time. The results are returned to the user at some later time. Often there is a delay ranging from a few minutes to several days. The facilities may be located several miles away from the school or may be relatively close by.

Obviously the programmer cannot personally interact with the computer. If there are errors in the program or changes the programmer wants to make then he must wait and resubmit his program. The two main characteristics of the batch system are the inability of the student to interact with the computer directly and the rather long time between the submission of the program and receiving the computer output.

"On-line" or interactive computer facilities consist of a computer and some sort of terminal (connected to the computer) such as a teletype-writer. The computer system may be a one terminal minicomputer or a time-shared system. A terminal usually serves as both input and output device. It must be located in an area where students can have access to it (such as a classroom). However, the computer may be several miles or even hundreds of miles away, connected to the terminal by way of a telephone hook-up. This set-up functions as if the computer were in the same room with the users. In the following paragraphs we shall examine batch and on-line systems in terms of their suitability for instructional purposes.

A key consideration in a batch system is the turn-around time. That is, how long does it take to get a program run. If it is a few minutes or less (which is standard at some universities) the system is much more suited to instructional use than if the turn-around time is "overnight" or several days.

Batch processing has been (still is) the major mode of computing at colleges and universities. Thus most computer programmers received their initial training in this mode, and most textbooks are written to facilitate this. A major problem with teaching computer programming on a batch system is that of keeping the students involved in constructive learning activities while they are waiting for the return of their programs. A "feedback" delay of several hours or days is discouraging to most students. Another problem might be the language available on the system. The most common programming languages available for batch systems and suitable for students are FORTRAN and COBOL. BASIC, which is currently the most popular for teaching students at the secondary level, is usually not available. (Note that this need not be the case. Batch processing BASIC systems do exist on some computers.)

Batch systems are not suitable for running computer assisted instruction (CAI) packages because of the need for immediate interaction. However, batch systems are quite suited to the running of most packaged programs. Most batch computer systems have extensive libraries of packaged programs.

If the facilities are "on-line" then all three of the instructional uses can be implemented to some extent. However, sophisticated CAI systems require quite a bit of auxiliary storage which may or may not be available on a given computer. Program construction requires very little in the way of size and can be implemented as long as a suitable language is available. Packaged programs can be used with minimum facilities also. There may be a few packaged programs which are too large for a very small computer, but there are still a large number which can be used.

As mentioned in the section on CAI the individual student use of a computer terminal is expensive. A somewhat limited time-shared facility (such as the Hewlett-Packard 2000 series) costs about $4000 per terminal per year. The lease-purchase of a more limited system, such as a PDP-8 with one terminal will cost in the range of $2000 to $3000 per year. Few schools can afford very many terminals. The lack of terminals creates problems in that students are rather slow at using the terminals. Only a limited number of students can use one terminal in a school day. One major task of teachers planning on implementing instructional use of computers is to assess their facilities and situation and come up with a solution to this problem.

Ideally, educators would determine what type of instructional computer facilities they needed on the basis of the tasks that needed to be performed. Realistically, instructional computer facilities are often provided somewhat independently of clearly defined needs. In such situations the nature and the extent of the facilities determine the type of instructional use which may be implemented rather than the use determining the facilities. Unfortunately, this situation will exist in many schools for some time to come.

The Computer is Coming--Eventually: The great majority of public schools (grades K-12) in the United States have no computer facilities available for instructional purposes. In some areas (large metropolitan areas, certain states) the situation at the secondary school level is better. In

Oregon, for example, about half of all secondary schools have some computer access. Typically this is provided via one or two timeshared (online) terminals in a school. That is, even in schools that do have computer access for instructional purposes, one terminal per 500-1000 students is common.

There are many reasons for the current paucity of computer facilities in the public schools. Major ones include high costs, lack of adequately trained teachers to use computer facilities, and lack of appropriate curriculum materials. We will discuss each of these briefly.

The issue of cost is a fundamental one. It was not too many years ago that cost of timeshared computer facilities was approximately \$20-\$25 per hour of usage. Currently a modern timeshared system designed for student usage could provide service at approximately \$1-\$3 per hour of student use. The cost of computer hardware has been decreasing rapidly for a number of years. Put differently, the price/performance ratio has decreased rapidly, and appears likely to continue to do so. It seems to be a common estimate that this ratio will decrease steadily during the next ten years. It seems to be a safe prediction that by ten years from now almost every secondary school will be able to afford to provide their students with computer facilities that will be judged excellent compared to what is available in the best of situations today.

The teacher training problem is one that will take many many years to overcome. Currently we have relatively little insight into the level of computer knowledge that is needed by teachers at various levels and in various disciplines. For example, the equivalent of a bachelor's degree in mathematics is considered desirable for the secondary school mathematics teacher. How much training in computer science does this teacher need if she is to teach computer programming or to supplement the conventional content of her course by the use of packaged programs?

Few states, school districts, or schools have comprehensive plans and/or goals for computer education for their teachers. Such a plan would take into consideration in-service programs, summer programs, sabbatical year programs, etc. It would also set desirable pre-service training criteria for new teachers. The absence of such plans in schools and districts that have invested heavily in computer hardware is particularly distressing. However, it is clear that considerable progress is being made. In-service courses are regularly offered in many of the larger population areas whose public schools have computers for instructional purposes. Pre-service mathematics education majors tend to take some computing courses, even when they are not required. A few schools, such as Illinois Institute of Technology and the University of Oregon have established computer science education master's degree programs. The more progressive graduate programs in education are making computer education courses available to their students.

Finally let us discuss curriculum materials. A good place to look is at the higher education level. Ten years ago there was little but some introductory computer programming texts, and books on numerical analysis. There was little idea as to what courses might be offered in computer science, or what the content should be. Now there are many well established degree programs in computer science. Comprehensive curriculum plans (for example, "Curriculum 68") have been developed and widely disseminated. New books at all levels have been published.

It seems likely that the same thing will happen at the public school level in years to come. As computer facilities and trained teachers become available to the schools, some degree of uniformity or stability in computers in the curriculum will develop. Existing books will be revised to take advantage of the computing facilities, and many new books will be written specifically for this market. In the area of packaged programs we can expect to see large numbers of programs available on the market to do most anything needed. These will be of a higher quality than that of those available today.

To conclude this section, we suggest that teachers and schools should look to the future as far as computers in education are concerned. The barriers are being overcome; the computer will arrive--eventually.

Non-Instructional Use of Computers

The non-instructional use of computers in schools is much more prevalent than instructional use. Although there are exceptions to this in higher education, there are few exceptions at the secondary or elementary levels. The main reason for this is that the computer has proven itself valuable for many non-instructional tasks; schools are more likely to acquire computer facilities for the staff (especially administrators) use than for students to use.

The non-instructional use of computers can be separated into two areas--administrative and ancillary. The administrative use is concerned with running the "business" end of the schools. The ancillary use is a catch-all category for those uses which are not clearly administrative nor instructional. We will discuss both categories (see Chapter VII for a detailed discussion of administrative uses of computers.)

Administrative Tasks: The use of the computer by the administrative division of a school system parallels the use in many modern businesses or industries. The computer is used primarily in the operation and management of the schools. That is, it is used to keep track of masses of detailed information, and to produce reports that are needed on a periodic basis. It does the normal tasks more quickly and efficiently and, therefore, is usually easy to cost justify. The following list presents some of the ways computers are put to use: payroll; accounts payable; generating reports for state, federal, and local agencies; test scoring; class scheduling; attendance records; personnel files; grade reporting; material inventory; and other normal accounting routines.

Currently this is the most common use of computers in the educational setting. Eventually one can expect that the computer will become more of a "management information system" and more inovative applications will be exploited. For example, the computer will be used to assist administrators to evaluate the instructional process in the schools by providing up-to-date results of testing programs, use of facilities and personnel, etc.

Having a wealth of information at his fingertips, the administrator will be able to make his decisions from a sound data base, and thus make better decisions regarding the planning, implementing, and evaluation of high quality education.

Ancillary Tasks: There are several uses of the computer which are not instructional (the student does not interact with the computer) and not administrative (an administrator is not involved). Examples of such ancillary uses include teachers using the computer to help in the management of instruction, counselors using the computer to assist students in making decisions, and librarians using the computer to assist students in research. In these cases the computer is used to support the personnel of a school who are in direct contact with the students. We will briefly discuss each of the three examples mentioned to provide more insight into the general area of ancillary uses of the computer.

CMI (Computer Managed Instruction) can be thought of as an automated teacher aide. (See Chapter VI for a detailed discussion of this topic.) There are a variety of CMI systems doing many different tasks in support of the teacher. In one system, the teacher uses it to store and retrieve information regarding some specific student or group of students. For example, the teacher may want to keep track of each student's progress in a reading program, what his test scores were for that year, what units he should study next, and how fast his progress has been. Along the same lines, it is possible to have a system correct and process a diagnostic test and print out a list of suggested goals for that student based on her test results, and print out a list of suggested references for her to study. In this age of stressing individual differences and individualized instruction the use of computer managed instruction is apt to grow very rapidly.

There are many new ideas regarding the use of computers to assist counselors. Two examples which are currently being tried out are 1) using the tremendous storage capacity and the speed at which information can be searched and sorted to assist counselors in helping a student find the most appropriate college or trade school or select a possible career area, and 2) using the computer to handle student records so that the counselor can keep track of a large number of students with a speed and precision which would otherwise no be possible. These two ideas by no means exhaust the uses currently being tried out, but they are representative of the efforts.

The use of the computer by librarians to assist students in research is a new area of computer application in the schools but is not without precedent in other fields. There are several rather large projects in the process of storing large amounts of information in computer files so that search for certain groups of items can be done by a computer program. One such project (the ERIC system--see the section on information retrieval in this Handbook) attempts to store information regarding educational articles published each month in such a way that one can ask for a list of all the pertinent literature on a specific topic (such as primary level individualized reading programs) and receive a listing of relevant publications and/or abstracts of them. This can provide a research service that cannot be duplicated by hand unless one has access to a rather large college or university library and a lot of help. Similar systems are possible for libraries, but it will be some time before it will happen.

Although these tasks are currently classified as ancillary, they are very closely related to instructional use. The only major difference is that currently the data received from the computer is not given to the student directly. It first goes to some staff member who in turn gives it to the student. In the future we may reclassify these tasks as instructional because the information will be given directly to the student.

The ancillary uses of computers is receiving more attention in the last couple of years and should be expected to grow rapidly. One reason for this is that the equipment necessary for many of the uses already exists in the administrative offices of the school. It is relatively easy in many cases to extend the user base to include the school's teachers. Extending the base to allow large numbers of students to use the system is more difficult.

Conclusion: Realistically, there are no clear lines dividing the many ways computers can be used in schools. The criteria used in separating one area from another was somewhat arbitrary, but convenient for purposes of explanation. The whole field forms more of a continum ranging from the most business-like administrative use to the students learning computer programming. A given task may lie in several areas at one time. For example, students could be tested with a computer on some aspects of the curriculum and the results given to 1) administrators so they could make long range plans regarding curriculum; 2) teachers so they could plan classroom activities to fit the class or student; 3) students so they could make more realistic decisions regarding their choice of classes; and 4) counselors so they may assist students in planning their post high school experiences. As the use of the computer becomes more widespread we can expect many similar applications to be developed and implemented.

As mentioned before, there is a lack of much valid research concerning the use of the computer for certain tasks. However, the lack of research in support of computers does not mean that the ideas are invalid. On the contrary, there is a large body of "opinion" that supports the use of computers. The problem seems to lie in the newness of computers in educational settings and the difficulty in generating objective research data concerning the results of using computers. Slowly this void will be filled, but, like many ideas in education, it takes a considerable amount of time.

The use of the computer in education will continue to grow and influence the public school curriculum and administration.

1. Allen F. and Bushnell, D., editors. The Computer in American Education. New York: John Wiley and Sons, 1967.

2. Curriculum 68: Recommendations for Academic Programs in Computer Science. Communications of the Association for Computing Machinery, Vol. 11, #3 (1968).

3. Spencer, Donald D. The Computer Goes to School. Association for Educational Data Systems (AEDS) Journal, Vol. 4, #1 (1970) pp 3-31.

SECTION D: WHY USE COMPUTERS IN EDUCATION?

David Moursund

Introduction This article could be a long, involved discussion of philosophical questions related to education as a whole, and the role computers can play in the entire process. Previous attempts by this author to write such an article have been "destroyed" by his colleagues. Thus, in this article we shall restrict our attention to a few of the essential ideas and "self-evident" truths. The result will be a relatively short discussion of a few of the key aspects of using computers in education.

Computers in Society At the current time, there are over 100,000 computers in existence in the United States. With the rapid development of mini-computers in the past few years this total number is, perhaps, a misleading figure. Mini-computers, costing in the range of $3,000-$30,000, are being mass-produced. These machines are small in price and physical size, but can carry out several hundred thousand arithmetic operations per second. The era of the "micro-computer" is just now beginning. A single large scale integrated circuit (one chip, costing well under $100) can be produced which contains all of the circuitry for the central processing unit for a computer. This same technology has produced the $50-$100 electronic desk and pocket calculators which are now readily available. We can expect to see many millions of micro-computers produced in the next decade. For example, it seems likely that every new car will contain one or more microcomputers, and that such circuitry may become common in TV tuners.

At the larger computer end of the scale we all accept the fact that computers are already widely used in business, government, and industry. Increased usage in all of these areas is to be expected. The use of computers in automation is really just beginning, and will have a profound impact upon our society. Zenith, for example, recently built an automated TV chassis production line in Chicago. Much of the operation is controlled by minicomputers. This one assembly line can produce about 2 million TV chassis per year, and uses about 1/20 the labor that would previously have been needed.

The main point is, computers are already widely used in the United States and are having a significant impact upon our society. But the proliferation of computers is just beginning! Much of the early development of computers was done in university research centers, and many early computers were used solely for research and instruction. Overall, however, the usage of computers in business and government has grown more rapidly than in education. The remaining sections of this article discuss the general issue of why one should (or shouldn't) use computers in education.

Computers in School Administration Education is a large business, consuming a significant fraction of the country's total productivity. In many ways the school administrator is like a businessman (perhaps one operating in a regulated monopoly?). Schools and school systems have income and expenses, budgets to balance, records to keep, reports to generate, etc. In general, computers are used in business and government when they are cost effective, taking into consideration the quality and quantity of work to be done. The same ideas should hold for the use of computers as an aid to school administrators.

The key concept here is that one is considering a change--either from not using a computer to using a computer, or a change of adding or deleting computer facilities. There are well defined, logical ways to attack this decision-making problem (whether to start using computers, use them more, use them less). One first gets a clear understanding of the information processing tasks to be performed. One studies whether they are currently being performed adequately, whether they should be performed, etc. One estimates as carefully as possible the costs of performing these tasks by alternative schemes. If it turns out that an information processing task which a school wants done can be done more cheaply by computer, then the school should probably move in that direction.

To conclude this section, it seems evident that more and more electronic data processing equipment will be used in the administrative aspects of running schools. This will range from $50 electronic desk calculators to $5,000 electronic accounting machines to $500,000 computer systems. The decision in each case should be based upon a study appropriate to size of the problem and the amount of money one is considering spending. The increasing information processing demands being placed upon administrators, plus the decreasing cost of electronic data processing equipment, point to an inevitable increase in the use of such equipment in education.

Teaching Using Computers One can consider computers as an educational media device. As such computers must compete with teachers, books, video recorders, movie projectors, tape recorders, etc. The competition is essentially in the area of cost effectiveness. For a given amount of money, we would like the instructional program to be as effective as possible.

It is worth noting that we do not yet have good insight (good instruments to measure) what makes an effective instructional program. This makes it difficult to predict whether or not a program will be improved by buying an extra movie projector, adopting a new textbook series, or adding a curriculum specialist. It also means that any change from the status quo (such as beginning to use a computer in the curriculum) is hard to "scientifically" justify. Of course, some research has been done on the effectiveness of computers as a teaching device. The area of computer assisted instruction has received considerable attention. In general, the results have suggested that CAI is about as effective as other teaching devices (such as teachers). In certain situations (such as with remedial students) CAI tends to be more effective than in other situations (such as with superior students). In any event, relatively little research has been done compared to what will be done in the future--and compared to what should be done before any large scale introduction of computers as a major instructional device in our schools.

Several key points are worth making, however. First, the cost of computer hardware continues to decrease (while teachers' salaries increase). The knowledge, understanding and computer software needed for making effective instructional use of computers continues to increase or improve. Research into the effectiveness of instructional use of computers continues to go on.

Added to the above points is the fact that computers are uniquely qualified for certain aspects of an instructional program. An interactive computer simulation program can be the basis for training an astronaut to pilot a rocket ship or a lunar module. A computer model of an environment, economy, or business can serve as the basis for interactive experimentation into aspects of the real world that are closed to most students. In many such instructional siutations, the cost of computer facilities is no longer the dominant factor. That is, the computer is cost effective in a number of special instructional situations.

To conclude this section, we repeat that a computer is a media device. A computer's unique characteristics make it more desirable or more suitable in some situations than other educational media devices. Over the long run one can expect that computers will prove to be cost effective in an increasing range of instructional situations. In particular, computer assisted instruction will eventually have a significant impact upon education, and computer managed instruction will probably become common.

Teaching About Computers In the earlier section on Computers in Society, we indicated that computers are widely used currently, and that their use will continue to expand rapidly. One generally accepted goal of education is to prepare students to cope with the current (and rapidly changing) real world. Computers are a fact of life! Thus, computers should be a topic of study in a modern educational system. We shall discuss briefly three possible aspects of teaching about computers.

Computer literacy, computer appreciation, or computer concepts, are all titles for a course of instruction designed to acquaint the student with capabilities, limitations, and implications of computers. Such courses have been available in some colleges and universities for nearly a decade. In the past three years a half-dozen good new textbooks, at the college freshman level, have been published. But such courses have been slow to reach into the public schools. They require trained teachers, texts, and equipment that tend not to be available in most schools. The 1972 report "Recommendations Regarding Computers in High School Education" prepared by the Conference Board of the Mathematical Sciences Committee on Computer Education recommended that a computer literacy course become part of the regular course offerings at the junior high school level.

Another aspect of teaching about computers is to teach the role, nature, and use of packaged (canned) computer programs. Quite a bit of what we teach (particularly in math and the sciences) is of a how-to-do-it nature. Thus we teach students to solve a certain type of equation, or to balance a proposed chemical reaction. The how-to-do-it parts of our educational system can, to a large extent, be stored in a computer program in a manner so that the computer can actually "do-it". Thus a computer can solve a complicated equation cheaper, faster, more accurately, than a person. (Some electronic desk calculators can compute the square root of a number as readily, and nearly as rapidly, as they can multiply two numbers.) Students need to learn to use such capability! Their education should build upon such machine capability--not compete against it.

Finally, we shall mention briefly the field of computer programming. With the continued proliferation of computers it is clear that some people will need to know how to program computers. At this stage of the development of the field of computer science, it isn't clear what percentage of students should learn how to write computer programs, or what degree of skill they should be expected to acquire. Some educators hold that computer programming is a suitable topic for the grade school, and that all students should learn to write programs. Others argue that a major sense of direction of computer science is to make a computer solve their problems without having to write programs. (Here the idea is that one uses packaged programs: someday it might be possible to just type in ones problem to a computer, and expect the computer to select and appropriate program for its solution, and solve the problem.) In any case, learning how to write computer programs seems to be one good way to gain some computer literacy.

Conclusion There is considerable justification to some current uses of computers in education, and to expanded use of computers. Uses of computers in school admininstration should be justified on a cost effectiveness basis. Similarly, use of computers as an educational media device should be justified mainly on a cost effectiveness basis. (An exception here is that computers are uniquely qualified for use in certain instructional situations.) The computer is having a profound impact upon our society, and upon almost all areas of human intellectual endeavor. Because of this, computers should be the object of study in our schools.

SECTION E: Abstracts of Related Articles for Chapter II

Division of Instructional Systems of the School District of Philadelphia
Laymen's Guide to the Use of Computers
The Association for Educational Data Systems, Washington, D.C., 64 pp.

Although the title of this publication does not indicate an emphasis on educational uses of computers, a brief examination of the sections reveals that section five, "Uses of Computers in Education," occupies a full two-thirds of the text material. As is stated in the preface:

> The emphasis of this guide is on instructional applications, showing how computers are now being used and can continue to be used to help in the educational process. It is designed to provide teachers, administrators, school board members, and parents with an introduction to some basic principles in data processing and to present an overview of the use of computers in instruction.

After a brief introduction dealing with reasons for including computers in the educational curriculum, several pages are devoted to the historical perspective and development of computers. Beginning with number systems, the abacus, and elementary calculators, the concept of computers is developed to an elementary understanding of the functions of I/O devices, the central processing unit, and the concept of storage.

Section III goes into more detail of the components of a computer and related peripheral equipment such as punched cards, magnetic tape, core storage, registers, arithmetic and logical operations, and a variety of output devices. Section IV describes the methods of writing and need for languages for a computer. Examples are given of machine language, assembly language, and FORTRAN.

Section V deals with the uses of computers in education and is broken down into twelve areas. Each area offers not only general goals, definitions, and objectives, but also specific examples and applications. The twelve areas considered are: 1) the computer as an administrative aid, 2) the computer as an aid to educational research, 3) time sharing, 4) the computer as a subject of instruction, 5) the computer as an instructional aid, 6) computer assisted instruction, 7) programmed instruction and its relation to CAI, 8) computer managed instruction, 9) simulation (including examples of four educational games: the Sumerian game, the Sierra Leone Development game, the Free Enterprise game, and the Consumer Game), 10) counseling, 11) other projects, and 12) the future of computers in education.

An eight page glossary of computer terminology is included.

Stewart G. Weimer

R. W. Hamming
Intellectual Implications of The Computer Revolution
American Mathematical Monthly V. 70, n 1, (1963)

The "energy crisis" is causing us to seriously consider the role of the automotive industry in our lives. We are finding that the effects are much more far-reaching than most of us imagined. Our culture lives and thinks in terms of automobiles. Prior to the "crisis", only a few intellectual prospectors bothered to explore these motorized ramifications.

Let's now turn to the computer for a moment. Is there a possibility that it has or will have a far greater influence in our lives than automobiles? The influence can undoubtedly be good, but is it all going to be good? What kind of crisis would (will?!) cause the man on the street to examine his life in light of the computer revolution? Educators had better examine some of these questions before they get their tails caught in some steamroller crisis. Computers can usher in an exciting new way to think and live or they can usher in a frightening new master-slave relationship. Hamming's article should encourage us to keep our tails well ahead of the steamroller!

"The Industrial Revolution effectively released man from being a beast of burden; the computer revolution will similarly release him from slavery to dull, repetitive routine."

"It is a common observation that a change of an order of magnitude in a technology produced fundamentally new effects."

"Automobiles are used at speeds about one order of magnitude faster than a horse and wagon. Each of these has produced whole new effects; ..."

"These are the bases of the computer revolution; at least six orders of magnitude increase in speed, at least three orders of magnitude decrease in cost, and an increase in reliability which makes practical computations involving billions of arithmetical operations."

"It is as if suddenly automobiles now cost two or three dollars, houses twenty to sixty dollars. And the changes in the computer technology are still going on!"

"Think, then, of the six orders of magnitude in speed and the three in cost, and the new effects they will produce."

Ron Edelman

Alspaugh, John W.
Utilization of Computing and Data Processing in Education
Clearinghouse, Vol. 43 (April 1969)
Pages 455-457

The use of computers and data processing equipment in education may be placed into three general categories: 1) computer-assisted instruction, 2) computing in support of instruction, and 3) computing and data processing in support of educational administration.

Computer-assisted instruction is usually considered to be an advanced form of programmed instruction using the computer for control. The special type of equipment needed for CAI is quite expensive, trained personnel are required to operate this equipment, and computer programs are expensive and time consuming to write. There are many engineering problems to be solved for the system to be efficient. There is also little evidence to indicate that CAI will reduce the number of required professional personnel. CAI is still in developmental stages and needs extensive research before it emerges from the laboratory and becomes a feasible approach to teaching students in the public schools.

Computing in support of instruction is concerned with the direct support of the teacher and students in the classroom but does not itself carry out any instructional process. The computing equipment can be used to grade, analyze, and record test results, to develop data banks of information about each student, to provide a laboratory device to solve many problems in mathematics and science, and to provide laboratory equipment for vocational courses.

Computing and data processing in support of educational administration is concerned with registration and reporting af grades, attendance accounting, payrolls, and financial accounting.

A thorough study of the clerical processes and records necessary for educational administration must be made before setting up a system for educational data processing. A systems study enables the district to eliminate duplication and provide more comprehensive and efficient systems. A computer allows a school to carry out routine information processing with faster speed and accuracy. Two obstacles have prevented more widespread use of computers in educational administrative data processing: 1) cost, and 2) lack of adequately prepared educators to plan and supervise the data processing activities.

There are three general approaches for obtaining computing and data processing services in a public school. They are: 1) contracting with an outside agency to provide computing time, programming support, and consulting services, 2) establishing their own computing and data processing center and 3) getting a group of schools to work together in setting up a regional data processing center or a statewide system coordinated by a state agency.

Hoeger

Blackwell, F. W.
The Probable State of Computer Technology by 1980, With Some Implications for Education
Journal of Educational Data Processing, Fall 1971/72

It is very unlikely that many of the innovative leaps in the computer industry development could have been predicted in advance. Similarily, although today's computer technology suffices for many needs in education today, changing computer technology of the 1970s will have real implications for education in the 1980s. The greater variety of systems, languages and data files will provide expanded uses in many facets of education and small general purpose electronic computers, mini computers, should be common place in another decade. By 1980 they should cost no more than $2000 as against $20,000 today.

It is easy to imagine a 1980 student spending part of each school day with a mini computer. It will be used for tutorial work, drills, listing and simulation. Teachers and administrators will find management and record keeping chores handled nicely by the mini computer.

Powerful software (language) is likely to develop. Software will handle very large memories and many levels of interrupts. The use of computers with simple languages will be routinely taught in secondary schools. The 1980s should see thousands of computer programs as readily available as library books are today. Many applications will be in education.

These advances will permit computers to be brought directly to administrators, teachers, and students. If costs go down there will be equal computer opportunity for all the nation's schools. "The infusion of all of these new ideas into the field of education is a very exciting prospect. Many political problems will have to be overcome, including the present lack of acceptance of computers at all levels of education, but willingness of those holding the purse strings to try new ideas in education, should insure that at least some of the new computer technology will be used."

Ray Korb

CODE, Ronald L.
<u>An Administrator's Guide to Computers</u>
College Management, October, 1972,
Pages 12-14

This excellent introductory article, although written primarily for college administrators, discusses issues and offers tips that can be of great help to any school administrator.

The article is divided into two sections. The first section offers an interesting classification of computer facilities. Code's categories are as follows:

(1) Ownership
  a. College owned, leased or rented
  b. Jointly owned with other colleges
  c. Not owned; services purchased
(2) Mode of operation
  a. Batch processing
  b. Interactive
  c. Both batch and interactive
(3) Control
  a. Managed for the entire college
  b. Under control of one department
(4) Application
  a. Administrative Data Processing
  b. Teaching programming
  c. Integrated with non-computer courses

The author explains each of these areas, and comments on each. This classification scheme can easily be modified to meet the needs of a public school administrator, and could prove a useful partial taxonomy of issues to be considered in joining or setting up a computer system.

The second part of the article suggests a coordinated plan for establishing and maintaining a viable and valuable computer program for the entire college. Code discusses such important issues as how to develop a procedure for faculty training, and how to involve faculty in a planning group.

The article is an introduction to the administration of computer programs designed to grow with, and serve, the entire school. It is not detailed nor technical, but rather intended to offer some important considerations, and suggest some directions.

Ken Bierly

Dieterich, D. J.
Magical, Mystical, Mechanical Schoolmaster; or, the Computer in the English Classroom
English Journal, Vol. 61, (December 1972)
Pages 1388-1395

There seem to be only two kinds of comments by teachers about CAI: very good or very bad. The people who swear by it note the individualization, the remedial or extra work it can branch to, the excitement children experience using terminals, even for drill and even for slow learners, and the freedom it gives the teacher from routine chores.

But the others see CAI as a threat to their jobs, something that will "destroy the teaching profession as presently practiced,...impose value systems often at variance with those traditionally revered", and a system in which all students respond identically and which can be used to gain "control of the mind of a nation." They wonder about teacher training in computers, how often the students will have personal contact with teachers, and interdependence and uniformity. They even wonder whether it will handicap or ignore poor students and whether "home computer consoles might have a marked effect on the quality of life."

Following Dieterich's article are 13 brief abstracts of ERIC Documents: 1) World Conference on Computer Education 2) The Computer in Education--EDEA 3) Bibliography of Programmed Instruction and Computer Assisted Instruction 4) Computers in the Classroom 5) The Possible Usefulness of Poetry Generation 6) Variable Modular Scheduling Via Computer, Developed by Stanford University and Educational Co-ordinates, Inc. 7) A Methodology to Achieve Secure Administration of English Comprehension Level Tests--Phase 1. Final Report 8) Drill and Practice in CAI Spelling: Word Ratings and Instructional Treatment. Project Interim Report Number 1 9) The Edison REsponsive Environment Learning System, or the Talking Typewriter Developed by Thomas A. Edison Laboratory, a Subsidiary of McGraw Edison Companv. 10) Cost and Performance of Computer Assisted Instruction for Education of Disadvantaged Children 11) Teacher's Handbook for CAI Courses--by Patrick Suppes and others. 12) Preparing for Computer Assisted Instruction and 13) The CAI Author/Instructor. An Introduction and Guide to the Independent Preparation of Computer Administered Instructional Materials in the Conversational Mode.

Marlyn J. Kern

JACKSON, Philip W.
The Teacher and the Machine
Horace Mann Lecture, 1967 at University
of Pittsburgh, University of Pittsburgh Press

This essay is divided into three assertions: 1) that changes in the teacher's work resulting from the growth of educational technology will not be as dramatic, and will not occur as rapidly, as many head-line-making predictions would have us believe; 2) that several of the educational benefits alleged to accompany technological change will either fail to materialize or, at best, will be mixed in benefits; 3) that although the expanded use of machines in the classroom poses some problems for educators, a more fundamental question concerns the extent to which a mechanistic ideology should be allowed to permeate our view of the educational process.

The author argues that many other mechanical teaching devices such as airborne television and instructional films have been heralded as dramatic break-throughs in education. Although it is foolhardy to counter one set of predictions with another, since future events have a way of eluding prior description, there is some reason to suspect that many of the bolder forecasts concerning technological change which involves computers in education will not be fulfilled.

The avowed goal of the machine promoters is not to displace teachers but rather to help them do their jobs more effectively. The essay discusses why teachers are not very open to technological change.

Given the pressures from government and industry urging educators to experiment with newer technological devices, there will inevitable be more widespread use of computers. But widespread adoption does not necessarily mean penetration into a significant number of classrooms. There are both strong and weak points in using computers as instructional aids. Much more concern will need to be given to the process by which people are treated mechanically and the educational problem of how to create and maintain a humane environment in our schools.

Ron Boys

Molnar, E.
Critical Issues in Computer-Based Learning
Educational Technology V. 11, n 8, (August 1971) pp. 60-64

Computer-based learning, as a classroom tool, is most widely found in colleges where 70 per cent of all college students are enrolled at schools having some computer instruction available. However, this percentage drops to 34 per cent when applied to secondary schools. Computer classes have traditionally been about the operation of computers. However, the use of computers as a tool in teaching quadrupled in the two years from 1968 to 1970 and continues to rise.

This rapid increase in computer-assisted learning approaches has led to a number of problems. Bernard Luskin is quoted as listing the seven most critical obstacles of CAI as being:

"1) availability of individuals with appropriate competent skills;
2) sufficient local funds for implementation;
3) sufficient funds for research and development;
4) attitudes of faculty;
5) lack of sufficient incentives to stimulate preparation of educational hardware;
6) poor documentation of educational software;
7) existence of a communications gap between educators and representatives of industry. (p. 61)"

Molnar feels that certain key issues must be resolved before CAI really reaches the take-off point in education. They are:

1. Is the computer a tool or a medium? As a tool, it improves the quality of education. As a medium, it allows education to reach more students.
2. Are we dealing with evolution or a revolution? The evolutionary approach would allow the student to be creative, while the revolutionary approach would require a modification of educational disciplines.
3. Do we centralize or decentralize computers? Centralization leads to larger computers with a cheaper time cost overall, while decentralization may necessitate mini-computers with limited capability.
4. Are costs benefits a myth or reality? Cost effectiveness is more difficult to substantiage in a culture where the telephone is seen as being more essential to the average person than is a computer terminal in a school building.
5. Can the federal government achieve a unified position in assisting CAI programs being implemented? Some programs get assistance while others don't with neglect and waste growing out of some programs. Furthermore, through the use of seed money, some programs are terminated before they really get off the ground. funding must not be limited to research but should include development and maintenance of programs.

In summary, are we going to allow the foregoing obstacles and issues to continue stagnation of an important component of the American education system, or are we going to utilize educational technology to meet the needs of our society?

Ray Dodson

Llana, Andres Jr.
COMPUTER TIME-SHARING: A DEVELOPING TECHNIQUE
AEDS JOURNAL, v2, n3, (March 1969), pp. 3-8

"In its simplest form, time-sharing is the simultaneous accessing of a single computer by many users, each of which is assigned a quantum of time within the central processor. This quantum of time is based upon several conditions and is subjected to continual swapping in and out of the processor until the particular user's needs are met."

Time-sharing was originally conceived as a simple means of lowering turn-around time for computational-type problems. It is thought of as being the first real "bridging the gap" between the user and the computer. The typical currently available time shared system can service about 16-64 simultaneous users. According to a recent U.S. Office of Education study, "1,000 simultaneous users would be more consistent with the needs of a typical large-city school system".

Allen Babcock developed a system called RUSH (remote use of shared hardware) whereby "They offer the user time-sharing with background batch-processing. In addition, there is a background-to-foreground capability in that a file developed and updated in batch enviroment may be queried in a time-shared environment." More recently G.E. has developed G.E.C.O.S. III, a system which "is a combination of time-sharing, background batch and foreground-to-background and direct-access programming. ...The present package offers the user an on-line time-shared operation, with batch processing, remote batch processing, and direct access." Such setups reduce the cost of time-shared computing to the user.

The educator is the largest potential market for future time-sharing services and at present is benefiting the least. BASIC, developed at Dartmouth College, has become a sort of standard for the industry. Recently, it has been learned that business has developed many business games, using BASIC, for use in their executive development programs. The Industrial College of the Armed Forces has spent years developing a complete set of management games which last from three to four days. Thus, for the educator, one trend seems to be toward simulations. Here a school can provide an ideal means of exploring a wide variety of problems, and by using a time-shared system, does not tie up an entire computer. CAI, by using a CRT, will make feasible new instructional materials founded upon new standards for instruction and course content.

"Certainly the opportunity is at hand for innovations in the use of computers in the environment of the educator. ...The educator has a stake in this developing technology. He must assert himself through active participation in its application to his environment."

Gerald Larer

Maloney, J. P. Jr.
Electronic Data Processing in Education
Journal of Educational Data Processing, Vol. 6, Winter, 1968-69
Pages 225-229

The major use of computers in elementary and secondary schools is in recordkeeping for purposes of administration. Schools today are moving away from pure recordkeeping and finding new ways for computers to handle the analysis of records to better prepare for the future. Computer analysis of areas such as instructional counseling, can highlight items that need immediate attention.

In class scheduling, schools have moved away from simple schedules to complex ones such as daily demand schedules, in which a student is rescheduled each day to meet his individual needs. Schools are experimenting with quarter schedules, year around schools, and modular scheduling.

With schools undergoing tremendous financial pressures there is a need to change from plain recordkeeping to recordkeeping with analysis. This is true for inventory problems, student records, and personnel analysis. Individual record files in the data base can now be subject to close analysis to staff schools better. Such files can contain information about curriculum content, staff experience, training, and need for new or updated training, facilities and their availability, and complete student records.

The next step in this trend is a total school information system. These new systems allow the administrator to have a comprehensive analysis of information available for important decisions. Allied with these systems is the so-called planned program budgeting system (PPBS), which tries to analyse possible alternative solutions and set a cost measurement on them. The computer can play a double role here: 1) it can help to develop new measurements; and 2) it is essential in processing the huge amounts of data that are created in testing, evaluation, and in general, material being generated by the education institution itself.

CAI (Computer Assisted Instruction) has volumes of material being written in its support, though this area is still very new. One important CAI potential lies in diagnosing student learning problems and suggesting alternative strategies. CMI, (Computer Managed Instruction) is also making progress to help individualize instructional content and sequence. Computer Concepts are being taught in the public schools as are computer applications. Some districts are moving forward by using computermobiles to reach students in outlying districts. The whole concept of vocational training in terms of data processing is being overhauled to prepare students to move along comfortably with computer technology and its improvement.

The computer manufacturers are currently well ahead of the educational thinking. Educators should try to reverse this situation.

Paul S. Ashdown

Meierhenry, Wesley C.
Computers in Education
Computers in the Classroom--317.3944 State Library, Salem, Oregon (1971)
Pages 141-162

This article provides a good overview of the field of computers in education. It begins with a short abstract, which is quoted below.

> "In his article which follows, Dr. Meierhenry shows a calm acceptance of the computer as a tool to be used by human beings for good or ill--a device with an impact on education which will force us to be precise about our goals and means in the classroom and will allow us, if we accept the challenge of its potential, to expand surprisingly our capacity to educate.
>
> As both teacher and educator, Dr. Meierhenry is concerned with the learner's need, as well as the logictics of the system. His paper ranges from the advantages and weaknesses of present instructional programs to the dangers of mechanical reliance on the computer. Well founded in today's structure, it looks to tomorrow with imaginative suggestions for concept development model-building in education."

The author hits upon many topics not generally covered in discussions of computers in education. Considerable emphasis is placed upon the role of computers in learning about the learner, and the learning process.

Oscar V. Evenson

Bork, Alfred
The Computer in Teaching-Ten Widely Believed Myths
SIGCUE Bulletin V. 7, n 4, (Oct. 1973) pp. 2-5

This article discusses 10 widely held misconceptions concerning the computer in teaching.

Myth #1: You must choose between direct and adjunct use of the computer.

Literature seems to stress that a choice must be made between adjunct use--where the student does his own programming, and direct use--where the student interacts with prepared programs. Actually, both moses can be accommodated in one system and can both be usefully applied to a variety of uses.

Myth #2: You must have massive equipment to use the computer in education:

Actually, many interesting applications can come from small mini computers, although they are more limited than larger models.

Myth #3: One language is much easier to learn than another.

Actually, the ease of learning a computer language depends more on the teaching methods employed than the specific language.

Myth #4: Computers will be widely used in education in present organizational structures of institutions.

Actually, computers are likely to revolutionalize the organization of schools.

Myth #5: At this time, your best buy for a terminal is a model 33 teletype.

Actually, educational users should never buy a model 33 teletype to day. They are much too noisy and slow, and maintenance costs eliminate much of the initial economy. A thermal printing terminal or graphic terminal is a better investment.

Myth #6: Computers are too expensive to use in teaching. This is a matter of bookkeeping. The stated costs of educational components depends largely on how the bookkeeping is done. Although it is hard to demonstrate, the author feels that computers are presently priced competitively with alternate teaching methods. In any case, while all other educational expenses are increasing, computers are becoming less expensive, thus more competitive with other teaching methods.

Myth #7: If we acquire a CAI language, that solves our problems.

Actually, the whole problem of an authoring system--incentives, facilities, personnel, testing and feedback procedures, text material--is more important than the language.

Myth #8: PLATO and MITRE are solving all the problems.

It would be unfortunate if the success of educational application of computers depended on these two large-scale projects. We need many diverse projects to explore all the possibilities.

Myth #9: Valid educational material can be developed without involving experienced teachers in the area.

Teaching is still teaching, on a computer or otherwise. The really effective educational materials are being developed by people involved in teaching, not just computer scientists or educational psychologists.

Myth #10: The computer used educationally uses only minor amounts of computer resources.

It is dangerous to plan computer uses in teaching under the assumption that minimal computer resources are required. Actually, many existing examples make great demands on computer core capacity and computer time.

David Malouf

Parker, John O.
Starting Computer Instruction? The Third Phase is the Most Difficult
Journal of Educational Data Processing, Winter 1967/68

The first phase of adapting to computerized instruction is that of finding small amounts of computer time for students.

During the second phase, the following situations arise: (a) learning to program the computer; (b) keeping communicatinn lines open with the company in order to obtain supplies and software; (c) inventing problems for students to use during the allotted time--during this phase a programming course was introduced to students.

After a few years primary emphasis moves to phase three, where and when to use the computer rather than how to use it. The problem is that one of providing expanded facilities for computer oriented instruction.

A formidable question emerges: once you have usable hardware and you know how to use it, what do you do with it?
(a) The computer may be used to teach students the concepts of computer programming.
(b) The computer may be used as a teaching aid within the regular coursework.
(c) The computer may be used by students working on individual projects.
(d) The computer may act as the teacher and the student is taught by action of the computer. This is commonly known as Computer Assisted Instruction (CAI).

In the Palo Alto school district, the general computer programming class consists of the following:

- (a) A general history
  - (1) discussion of hardware
  - (2) discussion of software
- (b) Machine language programming
- (c) Emphasis placed upon the order of operations in a symbolic language section
- (d) Problem solving in FORTRAN
  - (1) use of the language and monitoring system
  - (2) interactive techniques and looping
  - (3) program execution time
  - (4) rounding and cropping errors

In most cases the students are encouraged to select programming projects related to their other academic interests.

Palo Alto system has committed itself heavily to the premise that teachers must teach for deeper understanding even at some cost of time. Goals are reached through examples rather than lectures. The concepts taught should be those which are common to all computers, without undue emphasis on peculiarities of any specific machine. Experience with a real computer is an integral part of instruction, but the computer used for the course should remain the means to an end and not the reason for the instruction.

Ray Korb

Spencer, Donald D.
The Computer Goes to School
The Association for Educational Data Systems Journal. Vol. VI, No. 1, (Fall 1972), 3-30 and Vol. VI, No. 2 (Winter 1972), pp. 35-37.)

This is an excellent overview of computers in education and should help anyone who attempts to read this handbook. The concept of computers in education is broken down into six areas. After a brief introduction describing some of the present problems, the author discusses 1) recent developments in the field of computers, 2) the use of computers in the classroom, 3) a computer science curriculum for the secondary school, 4) the cost of a computer, 5) the necessity and extent of teacher training, and 6) some results to date.

The author discusses such recent developments as CDC's Star-100*, (a super-computer that can handle 100 million operations per second), multi-programming to allow one computer to handle several tasks simultaneously, the growth in the field of minicomputers, the development of simple languages, the increase in speed, and the overall decrease in cost per unit of computer power.

Section II lists several objectives of computer education in the secondary school: 1) to motivate students, 2) to allow for increased creativity and complexity in solving problems, 3) to better prepare the college-bound student, 4) to remove the mystery of computers, 5) to individualize instruction, 6) to provide teachers with a powerful tool, and 7) to provide a foundation for students who wish to continue in computer science. The use of computers in the classroom is then divided into six categories: as a tutor, a drill master, an experimenter, a student scheduler, a simulator, and a problem solver. Each area of discussion includes several concrete examples.

Several approaches to computer science education in schools are offered and course outlines for three different one-semester courses are presented. The section on the cost of a computer is divided into four parts: 1) the cost of a computer in regard to administration and educational use, 2) the cost of a time shared system, 3) the cost of a minicomputer, and 4) the cost per student.

Teacher training for computers in education should include general orientation, the influence of computers in society, computer solutions to a variety of problems, and methods of employing a computer in several discuplines. The sixth section, dealing with results to date, looks at fifteen different computer systems in fiteen different localities, briefly describing the system and its results.

S. Weimer

Knipe, Walter H.
The Computer At Grand Forks: Seeking the Individual
Educational Technology V. 13 n 8, (August 1973) pp. 44-46

When the school board at Grand Forks decided to individualize instruction, they first had the teachers establish goals of instruction for grades K-12. The second step was the writing of behavioral objectives which would serve as the nuclei of learning packages.

Each of these packages utilized a multi-media approach and offered to the learner a choice of learning experiences within the confines of the established behavioral objectives.

In 1971 the project began utilizing a PDP-12 with a 16K core and a 256K disc memory which would accommodate the following program languages: Basic Fortran, Focal, and PAL-D.

CAI

To date, over 200 math and science packages have been developed and revised. The average terminal time for these programs is 10-15 minutes but some programs accommodate groups of 3-4 students and thus one terminal can be used by 80-120 students in a given day.

The most popular of the programs involves simulation of real life experiments such as complex chain reactions in a chemical analysis of a given formula.

CMI

Criterion referenced tests in math are currently being utilized while similar tests are being developed in reading, science, social studies, and some vocational programs.

The individual responses on the test are read and scored by the computer and the teacher receives a list of names plus demonstrated weaknesses. The print-out also groups the results thereby allowing the teacher to group the students for help with common problems.

There is also an Instructional Information Service which consists of collected indexes of instructional materials in math and science. There is also a Professional Information Section which provides access to ERIC for the staff to develop further programs of instruction.

An excellent article of a system which seems to be accomplishing the sought after goals.

L. Shipley

Hess, Robert D. & Tenezakis, Maria D.
The Computer as a Socializing Agent: Some Socioaffective Outcomes of CAI
AV Communications Review. V. 21, (Fall, 1973) p. 311-325

The authors used students at a San Jose high school to determine whether students who used CAI or students who did not use CAI in their learning activities would have the most favorable attitudes towards the computer; towards the teachers; other media; the expertise of the computer vs. the teacher; and the trustworthiness of the computer vs. the trustworthiness of the teacher.

The results of the project (using semantic differentials) showed that students had a more favorable concept of the computer than of the teacher; that is, the students given CAI regarded the computer in more positive terms. For the non-CAI group, differences were even larger than the CAI group, indicating that these students had an even more favorable view of the computer as compared to the teacher.

The actual differences between the groups was in the images of the teacher rather than their images of the computer. The non-CAI group held a clearly less favorable image of the teacher than did the CAI group, whereas there was practically no difference between the two groups in terms of their ratings of the computer.

Both groups rated the computer more favorably than either textbooks or TV, on validity or accuracy of information. The computer was also seen as "faster." The computer outranked the teacher on items designed to elicit feelings of the students regarding the expertise of the computer vs. the teacher.

Differences observed and noted were believed to be the result from differences the students perceived in the learning situation in which they found themselves when working at the computer terminal and in class with the teacher.

Students believed the teacher was more responsive to student attempts to change "something", e.g., content or format, of their lessons.

The major reason for the more positive view of the computer was it's expertise in the field being studied. Students had a greater feeling of trust for the computer.

Implications of such research findings are that the teachers will have to develop new roles, for example, leaders in group projects; synthesizing information; and acting as a catalyst. The greatest work of teachers will be in the affective area, in subjects where the computer will be best able to teach the subject.

Jim Thiessen

CHAPTER 3

TEACHING ABOUT THE COMPUTER

by

Curtis Cook*

David Dempster

Mike Dunlap*

Herb Jolliff

R. Kohler

Bob Layton

Peter Moulton

David Moursund

Jim Muck

J. D. Smith

John Stedman

*Curtis Cook, Department of Computer Science, Oregon State University, and Mike Dunlap, Department of Computer Science, University of Oregon, served as group leaders and editors for this chapter.

SECTION A: OVERVIEW

This chapter contains material related to the computer as an object of study. It considers the major areas of computer literacy, computer programming, and computer science.

Section B deals with computer literacy. Because of the growing impact of the computer on society, computer literacy has become one of the most important topics in the study of the computer. It should be the first course in a high school or junior high school, to be followed by computer programming or computer science (for the student wishing further study). This section presents the need for computer literacy, basic course content, and some curriculum implications.

Sections C, D, E, F deal with computer programming. This section includes goals for computer programming; the CARDIAC computer and its ability to illustrate computing concepts and provide an introduction to programming; a discussion of what makes a good programming example, with some suggestions as to appropriate student handouts and teacher references, and an article on structured programming which shows the direction of computer language development.

Sections G, H, I deal with computer science. Computer science curriculum for the high school, small college, and university is suggested, followed by a rather difinitive discussion of computer science in high school.

Section J "Computing on a Shoestring" contains hints on starting a computing program with limited resources.

Section K is a set of abstracts of articles related to the chapter.

In general, this chapter pulls together much of the current thought related to teaching about computers.

SECTION B: COMPUTER LITERACY

by Mike Dunlap and David Moursund

Computers are a well established and rapidly growing "fact of life" in the United States, Western Europe, Japan, and a number of other countries. They affect all of us in our daily lives. For example, our telephone system is highly computerized; all large companies employ computers in their accounting and billing operations; all charge-card systems are computerized. Computers are essential to the operation of banks, savings and loan companies, and insurance companies. Airline, hotel, and motel reservation systems are computerized. The IRS uses computers to process federal tax returns, and the social security system makes extensive use of computers. Computer controlled machine tools and computer controlled "robots" for assembly work are beginning to have a significant impact upon industry. Often, in such cases, one is talking about the direct replacement of a human worker by a machine!

Who needs to know something about computers, and what do they need to know? What role should the schools grades 1-12 play in this general area? In the following pages we shall explore some possible answers to these questions. The reader should be aware that the questions are very difficult and that there is not universal agreement about the answers.

The "Competent" Adult Within any culture, there is a body of skills and knowledge which is generally expected of an adult. Some of these competencies seem to be universal: adults in all cultures are expected to understand and function within the social sanctions of their group, to communicate with each other in some way, and to attend to bodily needs. Others are culture-specific: the African pygmy must be able to move swiftly and quietly through the jungle, while the modern American is expected to be able to read, write, and do arithmetic. For each of these individuals, to possess the skills needed by the other would be as unnecessary as it would be unusual.

To single out any specific body of knowledge or skill, then, as a universally useful or desirable thing to know, is patently foolish. It is perhaps more reasonable to define such specific areas within a particular culture. Yet even here there are "twilight areas": most of us know, for example, one or more functionally illiterate adults who nevertheless lead useful and worth while lives in our own literacy-oriented country. Clearly, for any generalization about people there are exceptions. The existence of such exceptions should not, however, prevent us from trying to draw useful generalizations about the kinds of skills and knowledge that would be desirable for most people to have; and "literacy" is certainly such a skill.

The topic of computer literacy has been discussed by prominant computer scientists for at least the past decade. Their general conclusion is that it would be desirable for the adult population as a whole to have "computer literacy". The computer literate person is not intimidated by computers. He can function in a computerized world on a live-and-let-live basis. He has some knowledge and understanding of the capabilities, limitations, and implications of computers. He recognizes that people design and build computers, that people are responsible for the actions of computers. He sees the computer as a tool which can be used for good or for evil. To summarize, he understands that computers operate under the guidance of computer programs, and that computer programs are written by people.

Because computers are relatively new in our society, computer literacy is not widespread. The following (probably apocryphal) story of a situation which occured more than a dozen years ago, is still representative of the powers most people ascribe to computers. A graduate student at a large mid-western university carried out a research study and used a computer to analyze the data he gathered. He showed the results to his major professor; the professor rejected the results as being "impossible". The student then returned to the computing center and added a couple of statements to the computer program used to process his data. These statements caused the following sentence to be printed out at the bottom of the computer printout of the research results:

THE CDC 1604 COMPUTER SAYS THAT THE ABOVE RESULTS ARE CORRECT.

When the professor was shown this new computer printout he accepted the results of the research without further question!

All of us have read newspaper stories about people fighting with computers that repeatedly send bills for 0 dollars and 0 cents, and threaten court action unless the bill is paid immediately. Whose fault is this, and what can be done? The computer literate adult can cope with these and other products of computers in society.

The computer literate person is not overwhelmed by computers. When he receives a computer-printed personalized ad (for example, Readers Digest sends out tens of millions of these) he does not rush out and buy the proffered product. He checks his computer prepared monthly bank statement carefully (people type in the dollar amounts that are fed to the computer). He questions errors in billing. (A good example is a telephone bill. The telephone billing process is highly computerized, but people frequently get billed for calls they did not make).

An analogy is often drawn between computer literacy and "automobile literacy". Now-a-days students learn a great deal about automobiles and transportation by growing up in a world in which cars are an everyday mode of transportation. The point is made that in the early history of the topic (when automobiles were just beginning to be mass produced) there was a need for public education about the topic. This need went largely unsatisfied. Thus, for example, automobiles now polute our atmosphere to an untolerable extent, and have caused our cities to become concrete jungles. Perhaps some of this could have been avoided if proper attention had been given to the study of the capabilities, limitations, and social implications of automobiles.

The Long Term Future If we look 50-100 years into the future we can foresee a world in which computers are as common as automobiles are today. The preschool child may use a computerized information retrieval system to remind him that his favorite TV program is coming on, and to turn on and properly tune the set. The family TV set will also be an interactive computer-assisted instruction terminal. Adults and children alike may play computerized games, perhaps spending many hours a day interacting with a remotely located machine.

The computer will be integrated into all aspects of the public schools. (This assumes, of course, that schools are still recognizable in today's terms.) All teachers will know how to use CAI systems and computerized information retrieval systems; all schools will employ computer managed instruction. Thus all students will interact daily with computers, at home and at school. They will develop a great deal of computer literacy as a by-product of this interaction.

This will occur without benefit of any special instruction designed to teach computer literacy, and independently of whether or not computer programming is taught as an essential part of the school curriculum.

What Higher Education is Doing Before discussing current possible approaches to computer literacy via the schools grades 1-12 it is worth while to see what is going on in higher education today. We will use the University of Oregon as an example. Approximately seven years ago a computer scientist in the Mathematics Department of the University raised the issue of computer literacy, and proposed that the Mathematics Department offer a course designed to produce literacy. The course would contain no mathematics--indeed, it would contain no computer programming! Needless to say, the idea was rejected. A couple of years later a Computer Science Department was formed (by a split-off from Mathematics); a computer literacy course has been offered regularly for the past four years.

The issue of the appropriate content for a college level computer literacy course has been hotly contested from its very beginning. Among the supporters of computer literacy courses one finds two major camps: one favors inclusion of computer programming in the course, while the other is against it. As recently as four years ago there were almost no books suitable for a college freshman level computer literacy course, with or without programming. Early (non-programming) computer literacy courses at the University of Oregon either used no text, or used books of readings, or used other books which proved to be generally unsatisfactory. A change of texts occured almost every term.

The past three years have seen the publication of about a half dozen good books designed for use in computer literacy courses. These contain little or no computer programming. At the same time a number of computer science texts, designed for a semester or year-long freshman level course, have appeared. These books are strongly oriented towards computer programming but also contain a healthy dose of computer concepts, computer applications, computer implications, etc. Students studying such a course certainly gain computer literacy.

Schools of higher education, by and large, are not being overly successful in their attempts to produce computer literate graduates. Although appropriate courses are becoming increasingly available, many students choose not to take them. Of particular concern to the world of education are the attitudes of many college of education and teacher training program personnel. Many schools are still graduating mathematics education majors who have had no training in computer science. Many graduate education programs do not even encourage their students to take courses in the field ("other things are more important--and besides you know those courses are quite difficult"). An insignificant fraction of teachers preparing in the social studies fields receive training in the computer field. Most students in the social sciences are not being trained to discuss the social implications of computers, or to prepare their students to understand such critical issues as national data banks and invasion of privacy by computer surveillance. (Someday we may have a "scandal" involving invasion of privacy using national data banks that will make the Watergate seem like a Sunday school picnic.)

Computer Literacy in Grades 1-12 It is easy to see why little has happened so far in terms of computer literacy for the public school student. Most teachers know little or nothing about computers. Those teachers who do know something about computers are mainly mathematics or science teachers--hence unlikely to be particularly interested in including in their courses issues such as data banks and invasion of privacy.

The issue is compounded by a lack of appropriate texts and curriculum materials. Certinaly the definitive computer literature texts for junior high school and high school levels remain to be written.

Still another difficulty is the lack of adequate and/or appropriate computing facilities in most secondary schools. An important part of many computer literacy courses is having the students interact with a timeshared system, run a number of game playing and/or simulation programs, try out some CAI lessons, etc.

On the positive side we have the fact that a number of good movies and film strips on computing now exist. A significant fraction of the secondary schools in the United States now have some instructional computing facilities, and more and better instructionally oriented computer programs are becoming available in computer libraries. Some modern textbooks are beginning to contain units on computers. In many schools a reasonable number of teachers know something about computers (for example, perhaps all of the mathematics teachers know how to program in BASIC). All of this combines to provide a beginning, but much remains to be done.

The topics and ideas leading to computer literacy should be integrated into the curriculum grades 1-12. In the elementary school this would entail very little change. Students could be exposed to the concept of the computer as an information processing machine, and as a key element of communications systems such as a telephone system. A strong analogy between electronic desk calculator--human operator, and computer--computer program, could be developed. Students could learn that a computer allows automation of many of the activities that can be carried out by a man working with an electronic desk calculator. A visit to a computing center (including running some appropriate demonstration programs) and viewing some films might be the high points of the program. Perhaps most important at the current time at the elementary school level is that students not be fed misinformation. They get enough of this from the TV!

In junior high school the topic of computer literacy becomes more important. At this level the student should study the effect of the computer upon government business and industry. (The use of computers by the military is also worthy of special consideration.) Students should gain a familiarity with the capabilities and limitations of computers, and current uses and misuses. Most of this instruction should occur in the social studies courses taken by all students. However, the computer should be mentioned frequently in mathematics and science courses, since it is a fundamental tool in these fields.

It is at the high school level that the importance of computer literacy becomes critical. In the elementary and junior high schools the student is far from terminating his formal education, and the development of computer literacy can be a slow and carefully reasoned process. In high school, however, many students are on the verge of terminating their formal education. This is the last time they can be reached in a systematic manner. Thus a formal, required course designed to give computer literacy seems imperative. Such a course should probably be a semester in length. It should be taught by a teacher who is enthusiastic about the field of computers, and who has appropriate training and experience in that field. Following is a list of topics for such a course.

1) The student should understand what the computer is and what it does, and that the computer is not a magic black box or an answer machine. He should realize that computers make errors, what the source of those errors may be, and how they can be corrected. The student should become aware that the computer is a machine--a machine in the hands of PEOPLE.

2) The student should realize the value of computers in processing information for business and industry, with some study devoted to the applications of the computer in education, manufacturing, government, the military, and law enforcement.

3) The student should recognize some of the dangers associated with the computer. Generalized data banks such as credit files, medical files, and criminal history files should be studied along with their implications. Students should realize how the computer makes this information more dangerous or valuable because of the speed with which it is available. The people in control of computers can exert control over other people by using the information stored in the computer.

4) The student should become aware of some of the vocational implications of the computer. Here, information should be provided as to the effect that the computer is having on the nature of work, and what kinds of training the student might seek.

5) The student should have a perspective on where we have come as a result of the computer, or where we might be without it.

6) Finally, the future role of the computer for the betterment of our society should be considered in some detail.

The issue of whether one should teach some computing programming in this computer literacy course will not be resolved in this article. If adequate computer facilities exist, and the teacher is qualified to do so, then teaching some computer programming can enrich the course. Conversely, teaching too much programming will surely destroy it! A major problem in the next few years will be the variety of backgrounds that students will bring to the course. Some will have used computers for years, in their junior high schools. Others will have zero background plus considerable fear. Such a course will challenge the best of teachers!

As mentioned earlier, over a period of years the need for a specific computer literacy course in high schools will slowly disappear. The elementary school, junior high school and the home will gradually take over this aspect of computer education. The high school level material can be integrated into the social studies curriculum, and be handled by the social studies teachers.

Conclusion The computer is an important tool to our society. It increases our brain power, as our earlier machines increased our muscle power. But, as an extension of the mind, the computer is far more flexible and powerful than any of man's tools. As a result it is imperative that we prepare our children with the basic computer literacy that they will need in the world of tomorrow.

SECTION C:

GOALS IN TEACHING COMPUTER PROGRAMMING

Historically the computer tool was considered somewhat difficult to use; the first programmers were research scientists, most of whom had Ph.D.'s. Soon, however, graduate students were "allowed" to learn to write programs. By the early 1960's many colleges and universities offered computer programming courses that were open to undergraduates (the courses were generally numbered at the senior undergraduate level). It was about this same time that computer programming began to come into a few secondary schools, and it was soon demonstrated that even junior high school students could learn to write programs in a language such as FORTRAN.

The mid-1960's saw the development of BASIC and the spread of beginning programming courses to more and more universities, colleges, junior colleges, community colleges, data processing schools, etc. A large number of programming texts (in particular, FORTRAN texts) were published. By the end of the 1960's most schools were offering such a course at the freshman or sophomore level. In some reasonable sense the content of an introductory FORTRAN course at the college level was "standardized" by the rather standard content of most texts on the subject.

An introductory programming course can have a number of different goals. At the college level the two key goals tend to be:

1. To provide the student with adequate skill to use computers in his chosen field (business, engineering, etc.).
2. To prepare the student to go on the the next higher level course in computing.

At the secondary school level the introductory programming course does not fact the same restrictive goals. A list of some possible suitable goals for such a course is given below.* Note that a typical course will not have all of these goals.

1. To teach problem analysis and solution from a computer-oriented point of view.
2. To teach a subset of a compiler language such as BASIC, COBOL, FORTRAN, or PL/1, and to give the student skill in programming in the chosen language. It is important that the instruction be given in such a manner that it is not too dependent on a particular language.
3. To present the capabilities and limitations of computers (and perhaps of a particular programming language), and implications of the ready availability of computers.
4. To acquaint the student with the idea of machine and assembler language programming.
5. To teach the student how to use packaged ("canned") programs.
6. To give the student sufficient training in computing so that he can communicate with a programmer.
7. To introduce the student to the basic concepts, ideas and goals of computer science.

*From Dr. David Moursund's "Teaching BASIC and FORTRAN", Jan. 1973. (Unpublished manuscript).

8. To instill in the student certain attitudes toward computers.
9. To provide a combination of the above points which will prepare the student to go on to a more advanced course in computer science.

SECTION D

USES OF CARDIAC by Don Gallagher and Jim Muck

I. Introduction to CARDIAC

CARDIAC is an acronym for Cardboard Illustrative Aid to Computation. The CARDIAC illustrates the operation of a computer without actually being a computer. It is a practical aid to understanding computers and computer programming.

Many teachers who are looking for cheap, effective methods of presenting computer science concepts have overlooked CARDIAC. The CARDIAC is a simple computer, developed by Bell Telephone Laboratories. It is made of cardboard and has a set of ten machine language instructions. The student manipulates the parts of the CARDIAC, and in so doing gains an understanding of how the computer works.

The CARDIAC may be obtained free from Pacific Northwest Bell Telephone.

II. Uses of CARDIAC in Introductory Courses.

CARDIAC could be used in an introductory programming course. In this situation it could be used to lay groundwork for more advanced ideas. It can also be used as a motivation tool in encouraging flow charting and other basic concepts which will be discussed later in this article. Using CARDIAC as a tool will give the student experience he will find useful when he comes into direct contact with a computer facility.

CARDIAC could be used for a two to three day period in a computer literacy course as a means of introducing the basic parts of a computer, and illustrating the flow of activity within a computer. It should be noted, however, that only very simple programs should be written as this is probably the students first exposure to a machine language. Such use of CARDIAC should provide the student with a better understanding of basic computer concepts.

III. Uses of CARDIAC in Other Courses

Machine language programming:

CARDIAC may be used in any programming course as an introduction to machine language programming. With its set of ten operation codes and one hundred word memory, CARDIAC would be a valuable aid in an introduction to machine language programming for the high school student.

Assembly language programming:

An interpreter could be written in any available programming language which would interpret the CARDIAC machine code. This interpreter could then be used as a basis for offering a programming course in CARDIAC machine language at the high school level. (Writing an interpreter which executes machine code would be an excellent project for an advanced high school programming student). Another good project would be to build an assembler which produces machine code from assembly code. The interpreter could then execute that code. The development of an assembler could then lead to more advanced student's considerations of text editors and file handling routines.

Computer arithmetic:

A modification of CARDIAC to use another base arithmetic,(such as octal) would provide a means for introduction of computer arithmetic to the student. This again would aid the student in the understanding of computers as he obtains an appreciation of the way in which computers do arithmetic.

IV. Uses of CARDIAC to Illustrate Computer Concepts:

The CARDIAC is an excellent tool for illustrating many concepts of computers and computer science. The topics may vary from very elementary topics (parts of the computer, counters) through intermediate topics (bootstrapping, subroutines) to some quite advanced topics (sorting, stacks). The following discussion suggests possible uses of the CARDIAC in illustrating various concepts. An introductory course would cover the basic concepts; however, it is possible to introduce a few intermediate topics such as subroutines.

Basic Concepts:

Five Basic Parts of the Computer (, Memory, Arithmetic, and Control): The student becomes involved with the parts of the computer by manually moving data around and providing some control functions. He can see the interaction of the computer parts, the flow of data, and thereby gain an understanding of the role each segment of the computer plays in the process of computing.

Input/Output: By actually doing the input and output the student encounters the need for memory and the relationship between input/output and memory. The student is thereby able to distinguish between a memory location and the contents of a memory location.

Instructions and Data: The student soon becomes aware that within the computer there is no difference between instruction and data. Here the student begins to realize that the distinction between instructions and data is a control function of the computer and dependent on the placement of the program counter, which in CARDIAC is called the "bug".

Simple Programming: The CARDIAC provides a number of experiences in writing simple computer programs.

Algorithms and Flowcharting: The need for a step by step procedure to program the CARDIAC provides an excellent opportunity to introduce and work with the concepts of algorithms and flowcharting which are essential parts of a programming course.

Fetch and Execute Cycles: The student has the opportunity to observe first hand the control functions of the computer. The control cycles which fetch, decode and then execute an instruction are exposed. The special operations that result in control such as branching and testing data values can be observed in the control unit and through the actions of the program counter.

Counters: The concepts of programming repetition and looping can be observed by the student who may want to construct a counter as an example of both a control routine and a simple program.

Intermediate Concepts:

Fun is in store for the student who has progressed beyond the stage of basic concepts by examining the following concepts and questions.

Bootstrapping: How does the computer program actually get into the computer? The concepts in bootstrapping provide the student the answer to this question and at the same time show that the program which ultimately winds up as control information is itself read into the computer as data.

Loaders: A natural extension of the concept of bootstrapping is in the use of loaders. Once a loader is bootstrapped into the computer a smooth and easy access is available for other programs. One very important concept that appears here is in address modification, where the loader must have the capability of changing the location where it loads an instruction.

Since the CARDIAC does not have a multiplication or division command included in its instruction set, programs can be written using the concepts of repeated addition and subtraction. This leads to consideration of the following computer concepts.

Optimization: If the student is multiplying ten times two is it better for him to add 10 two times or 2 ten times? Does the program have to have the data presented in a certain order, or can the student find a method of testing the data and have the computer order it in the most efficient manner.

Overflow: What happens when the data becomes too large for the accumulator? What are the implications of not being able to store the overflow digit? The preservation of accuracy leads the student into considerations of double precision numbers and double precision multiplication.

Shifting: The shift command is necessary in double precision and illustrates one of the most powerful characteristics of computer registers. What happens to a number when it is shifted one space left or right?

Integer Division: How does the student decide when the division process will end and then what is done with the remainder?

Decimal Division: The use of storage and shift permit the student to continue a division beyond an integer result. What are the implications of decimal division? How could the student do double precision division?

Subroutines: A very crude subroutine jump and return mechanism is built into cell 99 of the CARDIAC from which the student can gain insight into the ideas of subroutines and subroutine package. Since only one subroutine is permitted at a time, the student may invent methods of saving the return address of the previous routine so that he may have a subroutine jump from another subroutine.

Advanced Concepts:

Using CARDIAC it is possible for students to consider more advanced concepts such as operations of the computer in memory space management or in the construction of higher level languages. Some of the following concepts and questions might be appropriate.

Arrays: The CARDIAC does not have provision for arrays; however the ingenious student can gain much insight into memory management through the implimentation of arrays. One of the questions to be encountered here is that of subscripting and the translation of the subscript into actual memory locations. The mappings that are involved in the use of the matrix will challenge even the top student.

Sorting: One of the better applications of CARDIAC programming is in sorting since the student must deal with input, output, counting, nested counting, arrays, subscripting, testing data, and possibly subroutines.

Stacks: The stack is a first in-last out type of list and is essential if the student is going to develop subroutine calls beyond the second level.

Simple Linked List: Memory allocation and use can be studied through the application of linked lists. Here the student provides both a data item and a pointer to the location of the next data item. The concept of dynamic storage allocation can be studied using the linked list which has application to keeping data in an ordered fashion.

V. Summary:

An acronym for Cardboard Illustrative Aid to Computation, CARDIAC provides a very practical aid to understanding of computers and computer programming. While we have seen that some very advanced computing concepts can be illustrated using CARDIAC probably its biggest asset is the fact that it is a very inexpensive method of providing an introduction to computer concepts and programming.

CARDIAC could be the basis for a programming course at the high school level at minimal cost.

CARDIAC is not a computer but it illustrates the operation of a computer and thus can be a very powerful aid to understanding whether you use it for a few hours as an introduction to computers or whether you use it for a complete course.

Perhaps the potential of CARDIAC has been overlooked. There are definitely more desirable ways of studying computers but probably none as simple or as inexpensive to use. Why not consider CARDIAC for use in your classes? You might be surprised at the results!

VI. References:

1. Saul Fingerman and David Hagelbarger, "An Instruction Manual for CARDIAC", Bell System Education, 1969.

2. Mike Dunlap "The Use of CARDIAC as a Pedagogical Tool in Computer Science", 1973. (Unpublished manuscript).

SECTION E

WHAT MAKES A GOOD PROGRAMMING EXAMPLE

by Robert Layton and David Moursund

Introduction: Courses in computing programming rely heavily on programming examples and exercises. Often the success of a course depends on the quality of the examples. A primary concern of teachers of computer programming at the secondary and community college levels is the lack of good examples which are suitable for use in class discussion or as exercises. A number of books are available which present and discuss suitable problems for solution on a computer. The fundamental difficulty, however, is that most general sources of problems do not take into consideration the particular needs of secondary school teachers. In the secondary schools one finds widely varying levels and lengths of courses, types of student prerequisites, depths of teacher knowledge, kinds of computer hardware and software, and degree of computer accessability. Thus most secondary school teachers of computer programming end up creating a personalized set of examples and exercises. This article discusses some of the features of a good example or exercise and presents a method for the systematic collection and presentation of examples and exercises.

Desirable Features in Examples and Exercises

It is said that "beauty is in the eyes of the beholder." To a large extent, this same is true of computer programming examples and exercises: what appeals to one student may be boring or confusing to another. Nevertheless, most good examples and exercises have some common features. A list of such features would include the following:

1. Relevance to the student interests.
2. Challenge-encourage the student to think.
3. Generalizability to a larger class of problems.
4. Expandability.
5. Appropriateness for computer solution.

Let us briefly consider each of these in turn.

Relevance - Examples and exercises should relate to the interests of the students. A seventh-grader is probably more interested in batting averages than in the Fibonacci sequence; a business oriented student would rather compute a compound interest table than a table of square root values. The teacher is faced with the challenge of addressing the problem or exercise to his specific student population. It is no secret that interested students tend to learn more than non-interested ones.

Challenge - The computer is a valuable tool in problem solving, and students should learn to use it as such. A computer programming course should teach coding as a means to problem solving, not as an end in itself. The exercises and problems should challenge the student's ingenuity and should encourage him to think of creative solutions. While trivial examples can be useful to illustrate the syntax of a programming language, more meaningful and challenging problems should lead the student up to--and beyond--the limit of his analytic skills. Challenge of this sort will stimulate the student to use his programming skills as well as to generalize them to other problem situations.

Generalizability - Wherever possible, an example or exercise should represent a broad class of computer applications. For example, a math oriented student might be asked to compute a table of values of an algebraic or trigonometric function. Such a problem may be used to illustrate the general concept of looping. Additionally, table construction is the key to the solution of many important problems in mathematics: non-linear equation solving in one unknown, finding the area under a curve, determining the extrema of a function, and solving differential equations are all examples of problems in which table-construction can be used. If the teacher recognizes the ways in which a technique generalizes to many sorts of problems, she can use this understanding explicitly to enhance the student's grasp of problem analysis.

Expandability - This characteristic is similar to generalizability, but has to do specifically with the broadening of a simple technique to solve progressively more complex problems. An example may be presented initially in a very elementary way, and gradually extended to more difficult situations. One might begin, for example, with the problem of finding the arithmetic mean of 5 numbers, and expand this to the general case of N numbers. The more open-ended an exercise is, the more likely it is to entice the students into investigating solutions on their own. Similarly, simple routines may be incorporated as subprograms in a larger problem over the course of a term's work; this enhances course coherency and provides some insight into program segmentation.

Appropriateness - In the real world computers are used for a broad variety of problems. Many problems involve large amounts of data manipulation (computation) and/or large amounts of input and output. Other problems involve nearly instant response to relatively complicated situations. Introductory courses in computer programming tend to be built around "trivial" examples. Thus, the student may be asked to write a program to compute the perimeter of a rectangle. It must be kept in mind that some learning can occur in the process of writing a computer program to solve a trivial problem. But solving a sequence of such problems makes little sense.

Of course the computer facilities available to the student as well as the typing (keypunching) skills of the student must be taken into consideration. Problems involving a modest amount of data preparation effort are desirable. Similarly, a problem whose solution involves 10-20 thousand arithmetic operations is to be desired over one involving a 10-20 million arithmetic operations. A problem involving 50 lines of output is to be desired over one involving 500 lines of output. Thus one must compromise between real world examples and what is expedient.

A Model for Problem Writeups

The systematic collection and classification of examples and exercises can serve two purposes. It can prove to be of value to the student in interpreting the problem and provide a means for documenting the exercise or example, for the use by other teachers.

The format presented here was developed by secondary school teachers in computer science summer institutes during the summers of 1970 and 1971. The format consists of two parts. One part provides guidance to the student in his development of the solution and the other is written for teachers as a reference.

One of the problems with a format for programming problems is that they do not always fit very well. For this reason the format given below should be viewed as a general guideline and should be modified to fit your needs.

Student Section

This model is designed to be handed to the student. It provides him with the information necessary to complete the problem.

PART I

(Student Section)

PROGRAM TITLE

(Title should be descriptive of the problem)

PROBLEM: Give a careful statement of the problem. This should be in a form so that, by itself, it could be handed out to students. Normally, the handout would include the data to be used in the problem and sample input/output (see below).

DISCUSSION (Narrative): Include a short discussion of the subject matter involved in the problem (for example, the subject matter might be physics or business). The basic programming concepts which are illustrated in the problem should be discussed. At this point the student should have an indication if he has the background to attack the problem. The discussion should show the relevance of the problem to the real world and to the subject of learning the program.

SAMPLE I/O: In this section the teacher will define the input data and what form the output data will take. At times this may take the form of suggesting what characteristics a good set of data (to adequately test the program) will take, and suggest the student create his own data. Many teachers use a standard data set, arranged so that it can be used for a wide variety of problems.

REFERENCES: A specific reference should be listed only if appropriate to students at a beginning level in computing, and if the reference is reasonably available to students. Otherwise this section should be omitted.

EXTENSIONS: Include here extensions on the problem to make the assignment bigger, harder, more useful, more general, or more challenging. The better student will want to include some of these in his program.

ADDITIONAL PROBLEMS: List additional problems of approximately the same level of difficulty whose solution involves essentially the same knowledge of a programming language. Alternatively use this section to list locations of where one can find additional, related problems.

TEACHER SECTION:

This model is designed to serve as a reference for the teacher. It gives a clear picture of the task and desired outcome.

PART II

(Teacher Section)

PROGRAM TITLE

(Descriptive of Problem)

Prerequisites:

a. Grade level or maturity level.
b. Computing (what language features are needed to solve the problem).
c. Subject matter (indicate the subject matter field in general and specific topics in that field).
d. Computing facilities (will one need particular hardware or software facilities not universally available).

Objectives: (Purpose, goals). The goals or purpose of the exercise or example might be stated as measurable behaviorable objectives or as teaching ideas as related to b, and c above, under prerequisites.

Discussion: In this section the writer attempts to communicate the knowledge and experience he has gained in previous use of the exercise. He may point out good use of the exercise, and how it fits into a particular course. The section may contain discussion on how to present the particular material under consideration.

Flowchart, Block Diagram, or Program Logic: Either a flowchart should be included, a block diagram, or discussion of the program logic, as seems most appropriate. Often one can tell the degree of difficulty of a programming exercise by a brief glance at a flowchart or a solution to it.

Sample Solutions: Generally a solution with sample output should be given. The solution should correspond to the program logic given above.

Additional Comments: (This section is optional). Generally these remarks would pertain to the solution given above.

The intent of the previous model is to enhance the communications between teacher-student and teacher-teacher. The systematic documentation of ones efforts will pay for itself in the long run.

Example: It seems appropriate at this time to present an example illustrating some of the ideas put forth in the preceding paragraphs. Note that the problem selected by a teacher must fit the environment (background of students).

Some assumptions have to be made prior to considering the following example. Assume a first or second course in programming is being taught and the students have taken at least an introductory course in algebra. The language available for use on the computer is BASIC, and the students have been introduced to subscripted variables and the "FOR..." statement. The example will be presented using the format for write-ups suggested in this paper.

STUDENT'S COPY

FIND THE LARGEST NUMBER

Problem: The problem is to develop a computer program to find the largest number in a set of integers. Write the program to use a maximum of one hundred integers. The program should print out the answer as follows:

The largest number is (answer)

Discussion: The problem, as presented here, is basically one in mathematics, but problems similar to this are encountered in the world of business. A company might request that their warehouse report which item last month, of several items,sold the most. If the inventory control was being performed by a computer, then a program similar to the one you will write could be implemented to provide the answer.

There are a number of algorithms which might be developed to solve the problem. One method would be to read the numbers into an array and then to compare the elements in the array to find the largest. Arrays are not required for this problem but will be useful on other related problems dealing with the a signed set of integers.

I/O: The set of integers to use for the problem are as follows: (25, 100, -43, 0, 84, 542, 67, 333, 89, 121). After you run the program using the above set, you may use a set of your own selection. Be sure to use zero, positive, and negative numbers.

Extensions: There are many interesting and useful problems that can be generated using the set of integers given above. Instead of the largest number, one might be interested in finding the smallest, or the median in the set. A problem frequently encountered is that of ordering a set of numbers in either ascending or descending order. Any of these problems are worthwhile and will present a challenge to the student interested in attempting a solution. Try one if you desire.

TEACHER'S COPY

FIND THE LARGEST NUMBER

Prerequisites:

a. The student should have at least an introductory course in algebra and one in computer science.
b. An exposure to subscripted variables, and the BASIC "FOR..." loop are desired. The problem can be programmed without subscripted variables or "FOR..." loops, but they are required on some of the extensions to the problems.
c. The problem is basically mathematical in nature, but can be related to many fields such as inventory control in the business field.

Objective:

The basic objective of the problem is to develop an algorithm to search through a set of numbers and find the largest. It also is intended to motivate the student in considering the more difficult extensions to the problem. The class can then pursue solutions to the extensions as separate assignments.

Discussion:

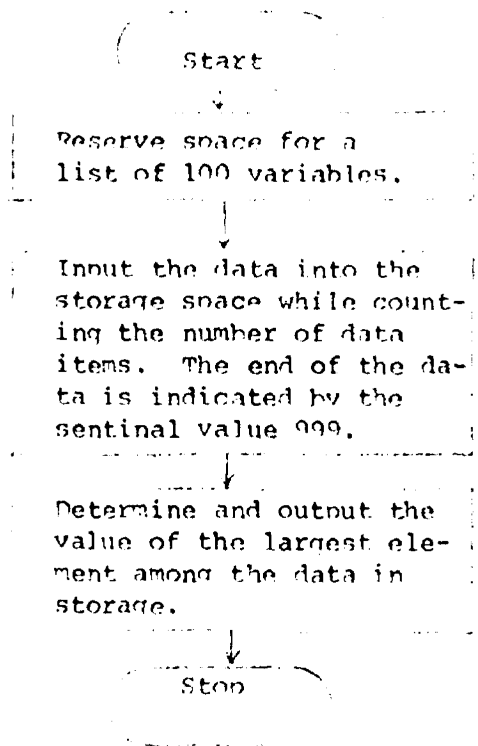
The problem has proved to be a highly motivational force in expanding the students knowledge in the areas of searching a set of numbers for specific properties. Initially the problem searches for the largest number but can be expanded to search for the smallest, find the median or the mode. The topic of sorting can then be considered by taking the same data set and sorting in either ascending or descending order. Related topics in statistics such as frequency counts, means, and standard deviations can become expansions of the basic problem.

The problem and its extensions create a learning environment in which the students' problem solving abilities can be enhanced at the same time his manipulative skills in the program language are further developed.

The point should be made with the students that the assigned problem, with its limited data set, is not a suitable computer program. If the numbers in the set were increased to 100,000, then a computer solution would be justified.

FLOWCHART AND CODING

The flowchart for one algorithm that may be used in solving the problem follows along with the coding. The problem was run on an interactive time-sharing system using BASIC.



```
10  REM COMPUTATION OF LARGEST ELEMENT IN A LIST
20  REM SENTINAL VALUE 999 INDICATES END OF DATA
30  REM N<=100 IS LIMIT ON LENGTH OF LIST
40  DIM X(100)
50  REM THIS STARTS THE INPUT SECTION
60  N=0
70  READ V
80  IF V=999 THEN 150
90  N=N+1
100 IF N>100 THEN 130
110 X(N)=V
120 GO TO 70
130 PRINT "ERROR!  TOO MUCH DATA."
140 GO TO 240
150 REM THIS BEGINS THE SELECTION OF LARGEST VALUE
160 B=X(1)
170 FOR I=2 TO N
180 IF B>=X(I) THEN 200
190 B=X(I)
200 NEXT I
210 PRINT "THE LARGEST VALUE IS"; B
220 DATA 25,100,-43,0,34,542,67,333,89,121
230 DATA 999
240 END
```

The point to be made concerning the problem presented here is that a simple problem can be expanded into a more meaningful, challenging problem for the student. The expansion into the more difficult problem provides the teacher with the opportunity to introduce new programming concepts. Usually the students will be motivated to take advantage of these new techniques.

Sources of Problems: There are many good sources of problems. The two most obvious ones are introductory computer programming textbooks, and books devoted to listing problems appropriate for computer solution. The references at the end of this article are all of the latter type.

Many good sources of problems exist around the teacher. Here are some places to look: Look at what you teach, and the textbooks you use, for ideas. Or, have the students suggest problems. Perhaps you could talk to other teachers to get ideas. What sorts of programs would be useful to you, your fellow teachers, or the school administrators? What sorts of programs might be of use to students in other courses, or to parents. The computer is a widely applicable tool--start looking for examples of applications and you will find them everywhere.

References for Computer Problems:

Barrodale, Roberts, & Ehle--Elementary Computer Applications in Science, Engineering and Business; John Wiley & Sons, 1971

Danver, Jean--Suggestions for Programs; Kiewit Computation Center, Dartmouth, 1970

Dorn, Bitter & Hector--Computer Applications for Calculus; Prindle, Weber & Schmidt, 1972

Forsythe, Organick & Plummer--Computer Science: Projects and Study Problems; John Wiley & Sons, 1973

Goldaker, Charles--Revised & Edited by Brook Firey--Computer Programs in Mathematics for Secondary Schools; Computing Center, Oregon State University, 1970.

Grumberger & Jeffray--Problems for Computer Solution; John Wiley & Sons, 1965
Maurer & Williams--A Collection of Programming Problems and Techniques; Prentice Hall, 1972
Nolan--Fortran IV, Computing and Applications; Addison-Wesley, 1971
Silver--Simplified Fortran IV Programming; Wiley and Sons, 1970
Steinbach--Programming Exercises for Problem Oriented Languages; Glencoe Press, 1969
Teague, Robert--Computer Problems for Fortran Solution; Canfield Press (San Francisco), 1972

SECTION

Stepwise Refinement, Structured Programming, and BASIC

by Peter Moulton

This section will discuss some recent developments in programming methods and the design of programming languages. These developments are concerned with writing programs and then determining that the programs work. They are gaining increasing attention of computer scientists and will certainly have considerable influence on the teaching of programming and on the programming languages we use.

A traditional method of teaching programming has been to present students with the statements of a programming language together with some examples and then set them to the task of forming combinations of statements with the hope that the resulting computation would provide a solution to the problem at hand. This skill at forming successful combinations of statements has been called the "art of programming". In some cases boxes and diamonds of flowcharts have been used to circumvent the syntactic details of statements and are often easier to understand. However, the search for the right combination remains. Certain standard patterns of statements dealing with standard problems have accumulated as programming techniques; but still programming has remained an art.

All too frequently this art degenerates into the practice of submitting an almost random sequence of statements to the computer in order to determine discrepencies between the resulting computation and the desired computation. (In some cases the desired computation itself is not really clear.) After altering the program to remove the most obvious errors, it is resubmitted in order to find the next set of errors. This might be called "programming by error", a frustrating, timeconsuming process yielding results which are usually mistrusted. However, beginning programmers frequently find having the computer do something wrong more rewarding than having it do nothing at all!

In the past few years increasing attention has been given to developing more organized and disciplined programming methods which might make programming more of a science than an art. The method currently receiving most attention is called <u>structured-programming</u> by E. W. Dijkstra (2), <u>step-wise refinement</u> by N. Wirth (2) or <u>top-down program development</u>. This method suggests that we bridge the gap between a problem and the desired program in a series of steps beginning first with a very general and short statement of what we want to do and then progressing through a series of expansions or refinements of the more general statements into sequences of more detailed statements. This process is continued until at the final step all of the statements are legitimate expressions in our programming language.

A very trivial example is to find the surface area and volume of a box. The first statement describing what we want to do is:

1) Determine the surface area and volume of a box. This is not a legitimate expression in most programming languages and must be expanded.

1a) Find dimensions of box

1b) Compute surface area and volume
1c) Print results

Again each of these three components must be expanded or refined.

1a) can be expanded into:

1aa) find length of box
1ab) find width of box
1ac) find height of box

1b) can be expanded into:

1ba) compute surface area
1bb) compute volume

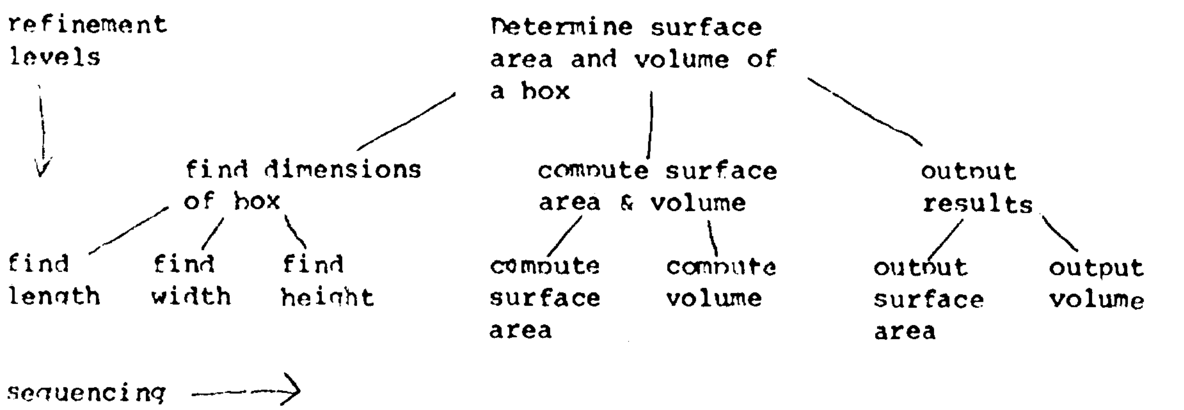
1c) can be expanded into:

1ca) output surface araa
1cb) output volume

At the next level each of these statements could probably be written in an expression of a programming language.

The brevity and generality of the first statement allows us to grasp the entire computation. As a refinement step expands a general statement into a sequence of detailed statements, the level of detail is limited so that the refinement can still be grasped as a whole. If the refinement is too detailed and results in a complex sequence of statements, it is difficult to convince ourselves that the refinement is equivalent to the more general statement. The rule is to proceed in steps sufficiently small to prevent getting lost.

As each of the more general statements is refined into a second level sequence of more detailed statements, the order of the second level statements is specified by the refinement. As each of these second level statements is being refined into a group of third level statements, not only is the sequencing within the third level group specified, but the sequencing between third level groups is specified by the sequencing of the second level statements. The ordering resulting from the step-wise refinement can be seen in a tree structure. For the example of the box:



The ordering of this structure is extremely important. For example it tells that any computation resulting from statements written as a refinement of "compute surface area and volume" must occur after that resulting from "find dimensions of box" and before that resulting from "output results." The importance of this ordering is that we may deal with the smaller problem of computing surface and volume and then know exactly where to place the corresponding statements, not getting this part of the refinement confused with other parts.

As the refinement process continues, the items of information or data being manipulated will become more explicit. In this way the specification of the data structures will develop in parallel with the program development. Deferring decisions specifying data structures as long as possible will help avoid awkward data structures not well suited to the computation.

Another example will be given after we have considered the other question: "how can we be confident that our programs work?"

As a program becomes larger and more complex, it becomes impossible to hold all of the parts in one's mind at any given time. This limitation is similar to that which brings the need for step-wise refinement methods and it is dealt with in a manner similar to step-wise refinement in reverse. One method allowing an awareness of the whole when dealing with any of the parts is to form groups of smaller parts according to some logical associations and then to think in terms of the functions of these larger units. The grouping of parts and the sequencing of the computation invoked by these groups is called the control structure of the program.

Ultimately our concern is not with the program statements themselves, but first with the computations invoked by the statements and then with the total computation consisting of the sequence of all of the computations invoked by the statements in a given execution of our program. It is seldom difficult to understand the action of a single statement taken by itself. The real difficulty is to understand the collective action of a group of statements. In order to understand the total computation, the control structure of the program must be understood. If our programming language provides features which enable us to clearly and easily grasp the control structure of a program, it will be easy to understand the total computation by examining the program. If we do not have a programming language with such features, then we must employ traces, intermediate dumps, and printouts in order to understand the total computation. For these reasons, the control structure features provided by programming languages have come under very close examination.

Dijkstra has suggested that the control structure of a programming language be based upon three basic types of control: concatenation, selection, and repetition.

Concatenation groups two or more statements into a single compound statement. The computation of the resulting compound statement is the computation of the first component statement followed by the computation of the second component statement and so on. The normal manner of indicating the concatenation of statements is to write one after the other, sometimes bracketing the group within special words such as begin, end.

Selection allows the computation to select between alternative computations according to some condition. Most commonly the condition is evaluated to true or false and a selection statement will take the form:

IF condition THEN s_1 ELSE s_2

where s_1 is a statement which will be executed if the condition is true and s_2 is a statement which will be executed if the condition is false. Statements s_1 and s_2 might be compound statements or selection or repetition statements. An example is

```
        IF A<B THEN BEGIN A=A+B
                          PRINT A
                    END
             ELSE B=B-1
```

Repetition allows the repeated execution of a statement (which might be a compound statement). In order to terminate the repetition, we can make its continuation dependent upon a condition. The form suggested by Dijkstra is

```
WHILE condition DO s
```

where s is a statement which will be repetitively executed as long as the condition is true. An example is

```
WHILE I>1 DO BEGIN FACT=FACT*I
                   I=I-1
             END
```

Obviously the repeated statement must alter the condition if interminable loops are to be avoided.

There are two very strong arguments for basing the control structure of a programming language upon these three statements. First they follow quite naturally from the process of step-wise refinement. Hopefully this can be seen to a limited extent in the example given below. If programmers are to be encouraged to use step-wise refinement methods, the refinement process must lead naturally into expressions of the programming language. If the final refinement step requires awkward or tedious translations into unnatural combinations of statements of the programming language, programmers will be encouraged to think ahead and work in terms of shorter or "more efficient" constructs of the programming language rather than follow the organization of step-wise refinement.

The second argument for using the control structure suggested by Dijkstra is that we can find methods to verify our programs. As Dijkstra has pointed out, program testing can only show the presence of errors, not the absence of errors. What we need are more direct methods to convince ourselves that our programs work. Dijkstra has given methods for proving assertions about programs constructed using concatenation, selection, and repetition.

It is significant that limiting ourselves to these three types of statements eliminates the GO TO statement common to most current programming languages. There has been considerable controversy over the use of the GO TO. (3) It has been shown that indescriminate use of the GO TO is a frequent source of difficulty in finding errors and that its inclusion in a programming language allows the possibility of control structures so complex that general program verification methods are impossible. On the other hand it has been shown that in many cases limitation to the control structures suggested by Dijkstra result in longer and less efficient programs than are possible with use of the GO TO. The use of the GO TO does not imply that our programs will be difficult to understand; its undisciplined use brings this about. At the same time limiting ourselves to Dijkstra's control structures does not mean our programs will be correct; it does mean we will have methods by which to analyze them and prove assertions about them. Using the method of step-wise refinement together with a language incorporating Dijkstra's control structures does allow us a more organized approach to programming with a greater chance of success.

Following is an example of program development by step-wise refinement. The programming language is a BASIC-like language incorporating the control structures suggested by Dijkstra. Although the program is still short and easily developed by less disciplined methods, it is hoped that it will illustrate step-wise refinement and, when compared with an equivalent BASIC program, demonstrate the more natural control structure.

Our task is to develop a general purpose drill program. The program is first to present instructions to the subject and then for each of a set of questions, present the question, accept the response, and keep score of the response. After the last question, the total score is to be computed and presented. Since the program is to be general purpose, the particular instructions to the subject and the set of questions and correct answers must be provided as inputs to the program rather than be coded into the program.

We have already described a procedure:

1) present instructions to subject
2) for each of the set of questions present the question, accept the response, and keep score
3) compute and present total score

If we had a programming language in which these were legitimate statements, our program would be complete. However, this is not the case; each of these statements must be refined. In order to present the instructions, they must first be input.

Step 1 can be refined into
 1a) input instructions
 1b) present instructions to subject
These two steps may then be translated into BASIC-like statements:

```
100 READ INSTRUCT$
200 PRINT INSTRUCT$
```

Where the first DATA statement is something such as

```
900 DATA ANSWER EACH QUESTION WITH ONE WORD.
```

Variable names longer than a letter and digit are used for the greatly increased clarity.

The second instuuction of the first level can be refined to:
 2a) input number of questions in drill
 2b) initialize current question number to 1
 2c) while there is a next question
  a) input question and answer
  b) present question
  c) accept response
  d) score response

The third instruction of the first level can be refined to:
 3a) compute total score
 3b) present total score to subject

All of these refinements with the exception of the last step of 2c, "score response" can be written as single BASIC statements.

There are many ways in which to refine this step depending upon the sophistication and complexity of our drill program. The most straightforward refinement might be:

<u>score response</u>

a) if response = answer then

let score = + 1

else present answer to subject

However we might want a retry feature incorporated as

<u>score response</u>

a) if response = answer then

let score = score + 2

else retry

Here we must refine the statement "retry". One possible refinement is:

<u>retry</u>

a) present "try again" to subject
b) input response
c) if response = answer then let score = score + 1 else present answer to subject

The important feature of step-wise refinement is that we can select any one of many refinements of "score response" without affecting any of the other parts of the program. We could also select a refinement with a "retry" and change refinements of "retry" without affecting the rest of "score response". This modularity allows us to work both at an overall level where details of the parts may be deferred and at a detailed level where relative independence of surrounding parts may be assumed.

Taking the refinement of "score response" with "retry" as given all of the steps may be easily written in a BASIC-like language to produce the following program.

```
010  READ INSTRUCT$
020  PRINT INSTRUCT$
030  READ NUMQUEST
040  LET QUESTNUM=1
050  LET SCORE=0
060  WHILE (QUESTNUM<=NUMQUEST) DO
070     BEGIN
080        READ QUEST$,ANSWR$
090        PRINT QUEST$
100        INPUT RESPON$
110        IF (RESPON$=ANSWR$) THEN
120           LET SCORE=SCORE+2
130        ELSE
140           BEGIN
150              PRINT "TRY AGAIN"
160              INPUT RESPON$
```

(Continued next page)

```
170          IF (RESPONS$=ANSWR$) THEN
180             LET SCORE=SCORE+1
190          ELSE
200            PRINT "THE ANSWER IS :" ,ANSWR$
210        END
220      LET QUESTNUM=QUESTNUM+1
230      END
240   LET SCORE=INT((SCORE/(2*NUMQUEST))*100+0.5)
250   PRINT "YOUR SCORE IS :"SCORE,"%"
260   END
270   DATA ANSWER EACH OF THE FOLLOWING QUESTIONS
280   DATA 3
290   DATA WHAT IS THE CAPITAL OF BRITISH COLUMBIA
300   DATA VICTORIA
310   DATA WHAT IS THE CAPITAL OF SASKATCHEWAN
320   DATA REGINA
330   DATA WHAT IS THE CAPITAL OF ALBERTA
340   DATA EDMONTON
350   END
```

Although a bit longer than the first example, this example is still somewhat trivial and several criticisms may be presented. Since the size of an example is limited and since we are concerned with methods for developing large and complex programs, any examples will be inadequate.

Perhaps the most significant criticism is that the example is too slick; everything works. In practice the method only offers some guidelines to help organize our programming efforts; it does not automate programming. Frequently on attempting to refine a general statement, a programmer will not know exactly how to proceed. The statement "find the best move" in a chess playing program does not have a clear easily found refinement. Frequently the attempt to further develop some step will bring about the need to alter some step earlier in the refinement; and the whole process must be backed up. This may require a considerable amount of rewriting and bookkeeping of names and sections of program. Some experimental interactive systems have been developed to aid a programmer with the clerical details of the step-wise development of programs. (4) For a fully developed non-trivial example of step-wise refinement see Wirth (2).

Once we have developed a program with these methods and a programming language designed for these methods, the product can be quite readable and easily understood. To illustrate this claim, consider the following example of an equivalent drill program as written by a typical BASIC programmer.

```
010   READ I$
020   PRINT I$
030   READ N
040   LET Q=1
050   LET S=0
060   IF Q>N THEN 210
070   READ Q1$,A$
080   PRINT Q1$
```

```
090  INPUT R$
100  IF R$<>A$ THEN 140
120  LET Q=Q+1
130  GO TO 60
140  PRINT "TRY AGAIN"
150  INPUT R$
160  IF R$<>A$ THEN 190
170  LET S=S+1
180  GO TO 120
190  PRINT "THE ANSWER IS:" A$
200  GO TO 120
210  LET S=INT((S/(2*N))*100+0.5)
220  PRINT "YOUR SCORE IS:" S;"%"
230  DATA ANSWER EACH OF THE FOLLOWING QUESTIONS
240  DATA 3
250  DATA WHAT IS THE CAPITAL OF BRITISH COLUMBIA
260  DATA VICTORIA
270  DATA WHAT IS THE CAPITAL OF SASKATCHEWAN
280  DATA REGINA
290  DATA WHAT IS THE CAPITAL OF ALBERTA
300  DATA EDMONTON
310  END
```

Although the abbreviated variable identifiers prohibit much of the description of the previous program, the real difficulty in understanding the BASIC program is in discovering its organization. In order to understand the computation which will be invoked, we must play "computer" and follow through each GO TO and IF-THEN. The larger organization of the program is not provided by the programming language.

The beginnings of BASIC date from the early 1960's. It is an easy language to learn; it has certainly received wide spread acceptance and has promoted the introduction of computers in many schools. However, it was designed without benefit of many of the recent advances in programming methodology.

Although criticisms are made about BASIC's limited identifiers or its file features or methods of input/output formatting, these are minor details when considering the teaching of programming methods and the development of well organized programs. For two reasons the fundamental shortcomings of BASIC are with its control structure. First its control structure does not promote well structured programs. Both the GO TO and the IF-THEN make use of line numbers which may be selected indiscriminately; there are no built-in checks or constraints. Programmers are required to think in terms of individual statements and are frequently tempted to take shortcuts in the name of efficiency (laziness?) which produce tangles of complex unreliable code.

Still a student programmer can be encouraged to develop well organized programs with step-wise refinement. Step-wise refinement is independent of any particular programming language. However, this presents the second fundamental shortcoming: methods such as step-wise refinement do not lead naturally into BASIC's control structure. Consider lines 170-200 of the first version of the drill program:

```
170  IF RESPON$=ANSWR$ THEN
180     LET SCORE=SCORE+1
190  ELSE
200     PRINT "THE ANSWER IS:",ANSWR$
```

A straightforward translation into BASIC would be

```
170 IF R$=A$ THEN 200
180  PRINT "THE ANSWER IS:", A$
190  GO TO 210
200  LET SCORE+SCORE+1
210  GO TO 150
```

which requires programming the ELSE case before the THEN case and leads to a redundant execution of a GO TO. Most programmers would quickly see other ways of coding this logic and would produce something such as the more efficient but less clear code of the lines 160-200 of the second version. A construct such as IF-THEN-ELSE--cannot be easily or naturally translated into a standard pattern of BASIC code.

What are the alternatives for a secondary school computer science curriculum at the present time? ---very few. Languages such as the one used for the first version of the drill program are being designed and implemented. Dijkstra's three basic statements are not the only ones for incorporating well structured features into programming languages and, in fact, have many limitations. Many alternatives are being examined. Languages have been implemented for experimental work, for limited instructional use in universities and for some production programming. (5). Some of these languages will soon be available on the small machines typically available in high schools. However, availability of a language is only a small part of the needs. Instructional materials, demonstration programs, in-service training are all essential and will be developed more slowly.

At the present time one might become aware of current developments in programming methods, make use of them where possible to encourage more organized programming habits and be aware of some of the shortcomings and pitfalls of languages such as BASIC.

References

1) Dijkstra, E.W., Notes on Structured Programming, EWD 249, Technical University, Eindhoven, Netherlands, 1969, reprinted in Structured Programming, Academic Press, N.Y. 1972
2) Wirth, N., Program Development by Step-wise Refinement, Communications of the ACM, 14 (April 1971).
3) Leavenworth, B.M., Programming with(out) the GO TO, SIGPLAN Notices of the ACM 7, 11 (November 1972)
4) Snowdon, R.A., Pearl: An Interactive System for the Preparation and validation of Structured Programs. SIGPLAN Notices of the ACM 7,3 (March 1972)
5) Wulf, W.A., Russell, D.B., and Haberman, A.N., BLISS: A Language for Systems Programming, Communications of the ACM 14 (December 1971)

SECTION G

COMPUTER SCIENCE IN THE HIGH SCHOOL CURRICULUM

by Jay Dee Smith

The intention of this section is to develop a notion of what computer science is and then construct a high school curriculum based on that notion. Discussion will be limited to secondary education.

Like most of mans' tools, the computer came into existence because of the demand--societies' demand for vast amounts of information that needed to be processed and the sciences' complex problems that needed solution. Today, the functioning of society and the development of the sciences are dependent on the computer. An individual cannot escape the effects of the computer.

As a general definition of a science, the following is offered: wherever there exist phenomena and the effects of these phenomena which are of interest to man, there can be a science to describe and explain these phenomena.(4) In particular, the discipline computer science, is the study of information representation and processing, the information processor (the computer) and the phenomena resulting from its use. As a field of study that is distinguishable from other disciplines, computer science is no more than 25 years old.

Just as the student of economics or mathematics starts with fundamental concepts and builds upon these concepts in later study as well as uses them in everyday life, the student of computer science should have some foundation upon which to expand his knowledge of computer science and, more simply, apply in everyday life.

As every educated person knows, his existence in today's society is based on his study in a wide range of disciplines. In particular, for the high school student, there are certain core requirements which have been predetermined to be necessary for his later existence. Because of society's dependence on the computer, every educated person should have a knowledge of the foundations of computer science, i.e., he should have as a part of his core requirements a computer literacy course.(3) Beyond computer literacy, a computer concepts and application course should be available to provide the student with programming skills he needs for personal use. What the student needs for personal use will depend heavily upon the individual person, the computing facilities available to him in later courses (or on the job), and his area of academic and vocational interest. A certain percentage of students will have the interest and the ability to go on to more advanced courses in programming and other aspects of computer science. This caliber of student would most likely continue in a computer science undergraduate education program.(7) Thus beyond the introductory concepts and applications course, a higher level course, advanced programming and machine structures, might be made available.

In addition to the three basic computer science courses suggested, modules integrating the computer into other disciplines should be created. For example, using the computer to compute the results of a physics lab from data frees the student to focus on the concepts of the experiment rather than becoming bogged down in a myriad of calculations.(3)

For the present, the above three computer science courses and modules to be developed are recommended as part of a well-established high school computer science program. A detailed description of the three courses will be found later in this section, however, at this time a discussion of the present state of computer science seems appropriate.

As stated earlier, computer science is an inquisitive child prodigy in a room of scientists well-disciplined in numerous fields. Computer science is a young, rapidly expanding field with its fundamental concepts and its role in education not too well defined. For most high schools computer facilities are usually limited. (See this chapter "Computing on a Shoestring", III.J) Good text and other instructional materials are not readily available. (See chapter 13). The teacher of computer science usually has limited formal instruction in the field but unlimited enthusiasm. (7) (See chapter 12).

It is not recommended that a school with no computer science department found one and immediately offer the three courses just to be on the bandwagon. Rather, the computer science program should be built carefully, with modest goals. As the program shows results and student interest grows, additional facilities and better instructional materials become available, and the teacher(s) gain experience and knowledge, a computer science program with a variety of goals can be offered. The computer science program suggested in this section offers courses that might be consistent with a program after 3-5 years of development.

Computer Science Courses

Following are the recommendations for the computer science program given earlier.

Course #	Course Title	Prerequisite	Core Requirement
C1	Computer Literacy	None	Yes
C2	Introduction to Computer Science	C1, High school math or Algebra 1	No
C3	Advanced Programming and Computer Structures	C2	No

Computer Literacy C1

Description: This course is designed to acquaint the average student of varied (or no) interests with the reality, nature, vocabulary, and the capabilities of the computer in current and future society. If the facilities are available, a programming language, e.g., BASIC, might be introduced. (See this chapter on CARDIAC,IIID) (A fundamental result of this course would be to dispel the idea or notion that the computer is a box with almost limitless capabilities and intelligence, and plant the truth that the computer is merely an incredibly fast machine that can be made adaptable to some of the needs of man.)

Major Areas of Study:

1. History and development of the digital computer.
2. Information processing - the elements of information processing using the computer.
   a. the notion of an algorithm - a finite sequence of operations or decisions which must be executed in a particular order to perform a job.
   b. the flowchart of the algorithm

c. The writing of the computer program from the flowchart using BASIC if the facilities are available.

3. The nature of the digital computer
   a. The basic requirements of an information processor, i.e., the taking in, processing, and the giving out of information, e.g., a telephone receptionist.
   b. The 5 main components of the digital computer, i.e., input, output, control, storage, and arithmetic and logical units (see this chapter on CARDIAC,III.D)
   c. The capacities and limitations of the digital computer--direct interaction with a computer facility using BASIC and/or preprogrammed packages would facilitate the teaching of these important concepts. (See chapter 4).
4. Wide sampling of the ways in which computers are used in our society with non-numeric as well as numeric application. The impact to those various uses on the individual should be made clear.(3)
5. Vocational opportunities available in computer science

Introduction to Computer Science C2

Description: This course is designed to acquaint those interested students with more refined problems solving skills, a more complete programming language, and the general field of computer science. (A student who makes continuous use of a computer will be faced with a wide variety of different software and hardware configurations. Specific details of how to use these systems vary widely. Thus, it is very desirable for the student to learn general ideas and principles, i.e., those concepts that are not dependent upon particular software or hardware.(7) It is also important that a student learn not just procedures for implementing a certain algorithm but also how to construct their own procedures.)

Major Areas of Study are outlined in Section I.

Advanced Programming and Machine Structures C3

Description: This course is designed to acquaint those students who have the ability and the interest to pursue computer science beyond the high school level with more detailed computer concepts, the nature of programming languages, and advanced programming techniques. (It is important that the student not learn just another programming language but be made aware of the structure of programming languages, i.e., the grammar, the construction of sentences in a language, the syntax, the rules of the language, and the semantics, the meaning of the syntactical structures. A structured language as ALGOL, MAD, or SNOBOL would give him an awareness of a natural communication with a language closely following the problem solving methods. Improved methods of problem solving are being developed; and advances have been made in programming language design and implementation. The methods and languages can be made available; it is important that they be incorporated into the school curriculum).(8)

Major areas of study

1. Computer concepts

a. Data representation - bits, bytes, words, double words; binary, octal, hexadecimal representations; binary and octal arithmetic

b. computer logic - introduction to Boolean algebra, AND, NAND, NOR, OR gates, flip flop, one bit memory unit for both registers and main storage.

c. computer structures - a review of the anatomy of the computer with an in depth look at the central processing unit, i.e., arithmetic-logic unit, control unit, registers, buses; a discussion of various I/O devices and interfacing

d. information flow and machine cycles - short machine language instruction set such as CARDIAC would facilitate the teaching of this section (see this chapter on the uses of CARDIAC, III.D).

2. Nature of programming languages - grammar, syntax, semantics; knowledge of a structured programming language with some programming experience. This step will require a great deal of time if proficiency in programming in a structured language is desired.
3. Advanced programming techniques - some suggestions are:
   a. program segmentation using subroutines
   b. file management, updating, searching methods
   c. use of storage printout or core dump for debugging purposes
   d. tracing techniques, i.e., tracing the steps followed during execution of a program and the printing out of data, addresses or instructions in core or the registers during each step. This is used for debugging purposes.

<u>Bibliography</u>

(1) Austing, Richard H. and Engel, Gerald L., "Computer Science Education in Small Colleges, a Report with Recommendations", presented at the ACM SIGCSE Second Symposium on Education in Computer Science, St. Louis, March, 1972.*

(2) "Curriculum 68", Communications of the A.C.M., Vol. 11, Number 3, March, 1968.

(3) "Recommendations Regarding Computers in High School Education", Conference Board of the Mathematical Sciences, Committee on Computer Education, 2100 Pennsylvania Ave., NW, Suite 834, Washington, D.C. 20037, April 1972.*

(4) Dorf, Richard C., Introduction to Computers and Computer Science, Boyd and Fraser Publishing Co, San Francisco, CA, 1972.

(5) "Computer Education for Teachers", American Federation of Information Processing Societies, Inc., 210 Summit Ave., Montvale, N.J. August 1970.*

(6) LaFrance, Dr. Jacques and Roth, Prof. R. Waldo, "Computer Science for Liberal Arts College", Wheaton College, Wheaton,IL, July 1972.

(7) Moursund, David, "Goals for Introductory Computer Programming Course", Department of CS, University of Oregon, Eugene, OR, June 1972.*

(8) Moulton, Peter, "Problem Solving and Computer Science in a School Curriculum", Department of CS, University of Oregon, Eugene, OR, February 1973.

*Abstracts on the above starred references can be found in the section of the handbook on Abstracts.

SECTION H

COMPUTER SCIENCE AT THE COLLEGE AND UNIVERSITY LEVEL

Many people have expended considerable time and effort to develop curriculum guidelines for the college and university. One widely known and significant study was conducted by the Association for Computing Machinery (ACM). This curriculum considered undergraduate and graduate programs and recommended courses of study leading to both the masters and doctors degrees in computer science. It appeared in the March 1968 issue of the Communications of the ACM, and has become known as "Curriculum 68". Some people felt, however, that "Curriculum 68" did not meet the needs of small colleges. So in the March 1973 issue of the Communications of the ACM an article appeared which considered the special needs of the small college in teaching computer science. Meanwhile, a group of people met to discuss and consider the special needs of the small liberal arts college. The result was the Wheaton College report. The Wheaton report was made available in July of 1972 and considered the college which had only minimal facilities, and was not interested in developing a computer science major. The Wheaton report is significant because it concentrates on providing computing as a basic support to other courses, rather than a study in its own right.

Abstracts of each of these reports appear at the end of this chapter. (See abstracts 3, 7 and 18 respectively.)

COMPUTER SCIENCE IN HIGH SCHOOL

"A Course Outline"

by

Jackson M. Dunlap

Introduction

Computer use at the high school level is increasing rapidly. Many schools already offer courses in computer literacy, computer programming and computer science. This paper is designed to illustrate an "ideal" computer science course for the high school level. It is written for teachers who are currently involved in computer instruction or about to enter this challenging field. The goal of the paper is to present one model of an exceptional high school computer science course in the hope that current programs will be measured against this guide.

It is the firm belief of the author that students who learn about computers in high school should be exposed to a comprehensive and broad based computer science course. Narrower approaches (computer programming) generally do not succeed in providing students with perspective about computers, computer use and computer function.

It should be noted that a course in computer science is not a study of computer programming. Programming is a definite part of computer science, but the major focus of computer science is a balanced approach to computers rather than an emphasis upon one part.

Overview of the Course

This course is designed to provide the student with a basic introduction to computer science. It considers the history of computers, a simplified look at computer programming, problem solving, problem logic with flowcharting, concepts of computer systems and architecture, survey of available computer languages, simple computer mathematics, elementary computer electronics, applications of the computer, the computer industry, and the future of computers.

Prerequisites include familiarity with the mathematical concept of a variable, and an ability to do basic computation (at the 9th grade basic math level). If a computer literacy course is available, it is desirable that the student take that course first. Such a course will deal more completely with the effect of computers upon society which is not the major objective of a computer science course.

This course should be taught so that the common average student will be successful. Some of the ideas are difficult when considered in their full generality; however, each can be handled in a manner acceptable to below-average students. Major ideas of computer science should be considered from an intuitive point of view and then expanded to an acceptable level of technical sophistication. The general level of this course should be satisfactory for an average tenth grade student.

Philosophy of Course - Problem Solving

The basic philosophy of this course centers on the concept of problem solving. Problem solving is seldom adequately taught in our schools. Yet problem solving is a basic survival skill. Students in a complex and integrated society need to know how to approach a problem, how to distinguish a solvable problem from an insolvable problem, how to break problems into fundamental parts, how to refine the subparts into a set of actions to be performed, and how to carry out this process in an orderly and systematic way.

One problem solving approach to this course is to consider basic problems of mankind. Man has faced some rather major questions in the development of society which serve as relevant examples of problems. The problems selected should be ones in which the computer has played an important role. Problems of this kind include war, manufacture of goods, the need for information, and the result of the information explosion. Each selected problem should be discussed, the solutions considered, and the role of the computer explored. In this way the student can see how problems affect mankind, how problem solving is approached, and the role of the computer in the solution. This process is characterized by dealing with a major problem, discussing how mankind solved it, and considering the part played by computers in the solution.

This philosophical structure to the course permits the subject matter to be intertwined in an interesting manner. As each problem is discussed, there are parts of problem solving, programming, computer systems, procedures, algorithms, and hardware involved. In this way the study of the computer is approached through the problems it has solved, rather than approached as independent units of study.

Level of the Course

A way to look at this course would be to consider it a one year introductory course with a horizontal development. Such a course would not have later topics dependent upon earlier information.

This method would not support advanced placement college programs. It would be of service to the wide group of students who will not become computer scientists and who may not even go to college. In this context it is like high school biology. The background gained in such a course can benefit the student in college, but the student is still required to begin his formal collegiate study at the standard freshman level.

In a one year introduction to computer science the instructor is free to examine the field without being under pressure to achieve a certain level of knowledge by the end of the course. The author believes that this is the best approach to computer science in the high school because it best provides for different groups of students.

The major drawback of this approach is that it does not provide the student who will major in computer science the strong, supporting background that is desirable. On the other hand, it does expose a wide variety of students to computer science and in the process may spark the interest of many future data processors; and at the same time it provides the budding computer scientist with some of the basic background that will ease his entry into a formal study of computer science.

The design recommended for this course is a free, flexible, and relaxed approach to the subject of computers and computer science. Such a design prohibits a rigid schedule. It requires the flexibility to follow areas of particular interests of students. Such an approach does require considerable skill and knowledge on the part of the teacher.

Goals

This course fits under three of the state goals defined by the Oregon Board of Education high school graduation standards. The three state goals are a) Personal Development, b) Social Responsibility, and c) Career Development.

The following goals are defined on four levels. The most general (denoted by roman numerals) are state wide goals. The system goals (denoted by capital letters) are student goals describing a major area of understanding subordinant to a state wide goal. Program goals (denoted by numerals) are related to a school or district program. And the course goals (denoted by lower case letters) are student oriented goals related to a specific course of study and are subordinant to the program goals.

I. Personal Development.
- A. Understand the basic scientific and technological processes.
  - 1. Understand the computer as a development of our technology and its function in a technological age.
    - a. The student knows basic problem solving techniques that can be applied to computer processing.
    - b. The student knows the elementary processes in design and construction of computer programs.
    - c. The student knows that different computer languages exist and that each is designed to fit a specific application.
    - d. The student knows the various configurations of computer systems, and their relative merits (batch, timeshare).
    - e. The student has an understanding of the nature of mathematics in computing.
    - f. The student knows about the technological development of the computer.

g. The student knows some of the expected developments in the future of computers.

II. Social Responsibility.

A. As a consumer of goods and services.

1. Understand the computer as a tool in our society.

a. The student will know some of the applications of the computer to areas like: business, industry, medicine, law, government, and education.

III. Career Development.

A. As an adult working within our society.

1. Understand the computer field as a place for employment, and the implications of computers in various vocations.

a. The student knows the basic employment categories in the computer industry.

b. The student knows the requirements for various types of work in the computer industry.

c. The student knows the effect of the computer upon currently available jobs.

d. The student will project the effect fo computers upon work in the future.

Instructional goals, behavioral objectives and performance indicators have not been defined here. These areas are more properly left to the individual teacher.

Course Topics

While this course should be integrated into problem oriented units, the basic topics permit independent discussion. The major areas to consider are: problem solving, computer programming, computer languages, systems concepts, computer mathematics, computer electronics, history of computers, applications of the computer, vocations in the computer industry, and computers in the future.

In each section that follows, the area is described in general terms and then considered in some greater detail. The discussion that follows attempts to make clear an area and its philosophical underpinnings.

Problem Solving

Problem Solving--The student should gain a feeling for the nature of an algorithm and a procedure. Both the flowchart and the English language procedure should be used to increase student problem solving abilities. Problem solving supports the programming sections of the course and increases the skill of the student in solving general classes of problems.

Problem solving is a key element of this course. It is the logical framework around which the course is built. As such, it requires significant amounts of class time and will be one of the most difficult topics to teach. One reason for the difficulty of problem solving is that students frequently come into computer science poorly prepared to logically approach problems. Fortunately, students can learn problem solving if they first realize that each major idea of component of a problem must be isolated and dealt with individually. Once a student has this basic skill it is easy for him to solve more advanced problems.

Problem solving is a basic part of computer programming. It should be introduced prior to any programming and used as a tool throughout the programming portions of the course. A tool to illuminate problem solving is the flowchart, which lays out the logical elements of a problem in a graphical form. A second tool in problem solving is the stepwise refinement.[1] In stepwise refinement the problem is thought of as large independent blocks. Each of those blocks is subdivided into smaller blocks, (English language procedures) until the problem is defined precisely enough that it can be solved with a number of well defined and discrete actions.[2]

It should be noted that flowcharting and stepwise refinement are general procedures. They are applied to programming but are general enough to be applied to nearly any problem.

Flowcharts must be done CAREFULLY. It is important to realize that when computer code is written into the boxes of a flowchart, the chart loses most of its meaning. Flowcharts contain English language expressions; that is, flowcharts contain boxes each of which has a meaning, and the contents of the flowchart box is an English language description of the action to be performed. Careful attention to this point permits the student to bridge the gap between computer and human language definition of the problem.

It should be noted that flowcharting, and stepwise refinement work together in the solution of problems. The flowchart being the graphical representation of the problem and the stepwise refinement a method of problem definition.

Computer Programming

Computer Programming--Programming should include one simple computer language. Reasonable programming topics include, the concept of a variable and its relationship to the memory, the process of program documentation, its importance to the programmer and to the industry, input/output, simple mathematical calculation, a discussion of control statements and, their function within a program (LIKE: IF, GOTO, AND looping mechanisms), strings, basic concepts of loops and counters, and simple one-dimensional tables.

---

1 Wirth, Niklaus; "Program Development by Stepwise Refinement," Communications of the ACM, April 1971, Volume 14, Number 4.

2 Stepwise refinement is an important concept in computer program design and should be a part of any respectable computer science course.

The programming aspects of the course should not exceed one-fourth to one-third of the total time (including lab work in programming). The programming portion of the course is designed to expose the student to programming, not to make him a programmer. The material learned here should help the student to better understand the computer.

Experience working with students at this level indicates that most students can learn computer programming through the one-dimensional table. After this point the difficulty of the material increases rapidly.

A teacher is wise to set the limit of programming to the one-dimensional table. Topics beyond this level should be handled in more advanced courses, devoted to programming.

The student who is capable of absorbing considerably more programming than is suggested here should be given a manual and permitted to devote any extra time he has to programming.

Sample programs, exercises and examples should develop a balanced approach to computer programming with a (50-50) mix of string (alphabetic) and numeric types of processing. Strings are important because they require less mathematical sophistication, and illustrate the idea that computers do more than process numbers. The key idea here is to move away from math oriented problems and give the non-math student a chance to perform in programming.

Some examples of programs that can be developed with strings include: the form letter, billing for an electric company, drawing pictures, CAI programs, labeling of output, and designation of input.

The study of programming is not complete without a discussion of structured programming. It is an important modern concept in computer programming and has significant relation to human problem solving.[3]

Computer Languages

Computer Languages--Students should be exposed to computer languages including: translation of languages, compilers, interpreters, and assemblers.

One interesting topic in computer science is a discussion of various computer programming languages. Today there are a proliferation of computer languages, each with a different purpose. Of the more than one hundred computer languages, several have specific purposes and value. Such languages include FORTRAN, APL, COBOL, BASIC, ALGOL, BLISS, SNOBOL, PASCAL, and PL/1. Students should be aware that computer languages are translated by computer programs and that there are both machine and higher level languages.

3 Dahl, O. J., Dijkstra, E. W., Hoare, C. A. R., Structured Programming, Academic Press, New York, 1972.

System Concepts

System Concepts--Students should study: major components of the computer, the basic architecture of the computer, software including concepts behind timeshare and batch, and the functional relationships between the parts of a computer system.

The computer is more than a central processing unit. The computer is a system of components that interrelate in interesting and complex ways. The study of the computer as a system is an important topic which provides the student with a firm understanding of the nature of computer processing.

Computer Mathematics

Computer Mathematics--The student should gain an understanding of basic computer mathematics including: binary, octal, decimal notation· why binary and octal are used in computer systems; an intuitive view of precision and accuracy including the effects of round off errors, and that methods exist for increasing the number of significant digits available in the computer.

The mathematics of this course should be kept as simple as possible. It is very undesirable to deal with advanced mathematics or even topics common in algebra. It is entirely possible to consider the topics mentioned above from an intuitive and numeric basis without any sophisticated mathematics.

Computer Electronics

Computer Electronics--Students should study core memories, why core memories are used, newer types of memory devices (i.e., solid state), basic logic and logic gates (AND, OR, NOT, INVERTOR, FLIPFLOP, HALF-ADDER, FULL-ADDER), the basic circuitry necessary to construct gates using transistors, a study of logic represented with primative gates, changes brought on by miniturization of electronic components, integrated circuit devices, and circuit manufacturing.

The computer is an electronic device. Part of the mystery of the computer is veiled by the basic function of its electronics. When students realize that the computer is constructed from discrete functional components they also begin to understand that the computer is not magic. These topics provide basic and fundamental understanding of how the computer works.

History of Computers

History of Computers--Students should study the history of computers including: the effects of computers on problem solving, the development of computing devices, the generations of the computer, and the effects of various problems on computer development.

An interesting way to deal with the history of computers is to present a basic problem, consider the problem, analyze solutions to the problem, and the technological changes brought in computers by the problem. One example of this process is to examine the events leading to the development of Pascal's adding machines, Babbages Difference Engine, or the punched card.

A study of history inter-ties various developments of the computer and provides a platform for consideration of computer applications.

Applications of the Computer

Applications of the Computer--Students should consider: applications of computers to education, government, law enforcement, medicine, process control experimental design and control, models and simulation.

It is extremely important for the student to gain an understanding of how various industries use the computer, what effect computers have upon that business, and what the computer might do in the future for that business.

Vocations in Computers

Vocations in the Computer Industry--Students should explore jobs in and associated with the computer industry, including: keypunch, control clerk, control supervisor, computer maintenance, field engineer, customer engineer, DP manager, computer operator, shift supervisor, operations manager, computer sales, programming computer applications, systems programming, systems analysis, computer design, and computer manufacturing.

Implications of employment in the computer industry is important. Many of the jobs in the future will be either entirely or partially related to the computer.

Computers in the Future

Computers in the Future--The student should consider the expected impact of computers upon education, government, law enforcement, medicine, automation, research and society.

Teacher Skills

It is evident that this course requires considerable formal background and experience with computers. In the future teachers of computer science, like teachers of biology, will have a level of training equivalent to an undergraduate degree in the subject field. Unfortunately, few teachers currently have the qualifications necessary to teach this course.

A teacher who does not have the full qualifications for this course can do two things. First, he should go ahead and teach his current courses. As the courses are taught they should be compared with this model. As the teacher notes area of departure, he should zero in on a few topics of interest and over time improve his knowledge of those areas. After a few years of this activity the teacher will increase his overall knowledge and come closer to the set of desired qualifications.

A second thing the teacher can do is return to school. Several fine universities offer summer programs in computer science for teachers. Some of these programs lend to a master of arts in teaching computer science. This type of training, which does not interfere with the standard teaching year, will increase the qualifications and proficiency of teachers very rapidly.

A very modestly qualified teacher may desire increasing the actual balance of programming to computer science within his course. While computer courses at the high school level should not become programming courses, it is respectable to increase the balance of programming while the teacher gains wider experience. Once the teacher feels more comfortable with computer science he should begin to look at various areas of the course and make adjustments away from programming.

In time, and with some effort, the person who is currently involved in computer instruction can become a master in the subject.

CONCLUSION

The outline presented above is a radical departure from typical high school computing courses. Its emphasis is upon understanding, exposure, and integration of ideas. It is, indeed, a difficult course to teach in its full generality, but with a talented teacher and the proper circumstances, this course could become one of the most significant in the student's school background.

Bibliography

Adams, J. Mack, and Moon, Robert; An Introduction to Computer Science, Scott, Foresman and Company, Illinois, 1970

Albrecht, Robert L., Lindhert, Eric, and Mara, Walter; Computer Methods in Mathematics, Addison-Wesley Publishing Company, Menlo Park, CA, 1969

Anderson, Decima M.; Computer Programming: Fortran IV, Appleton-Century-Crofts Educational Division Merideth Corporation, New York, 1966

Ball, Marion J.; What is a Computer?, Houghton Mifflin Company, 1972

Corliss, William R.; Computers, United States Atomic Energy Commission/Division of Technical Information, Oak Ridge, Tennesee, 1967

Crowley, Thomas H.; Understanding Computers, McGraw-Hill Book Company, San Francisco, 1967

Dahl, O. J., Dijkstra, E. W., Hoare, C. A. R.; Structured Programming; Academic Press, New York, 1972

Dorf, Richard C.; Introduction to Computers and Computer Science, Boyd & Fraser Publishing Company, San Francisco, CA, 1972

Dymax; My Computer Likes Me *When I Speak in BASIC, Dymax, Menlo Park, CA, 1972

General Electric; You/and the Computer

Gottfried, Byron S.; Programming with Fortran IV, Quantum Publishers, Inc., New York, 1972

Hamming, Richard W.; Computers and Society, McGraw-Hill Book Company, San Francisco, 1972

Hull, T. E. and Dan, D. D. F.; Introduction to Computers and Problem Solving, Addison, Wesley (Canada) Ltd., Menlo Park, CA, 1969

IBM Corporation; More About Computers, Armonk, New York, 1969

Johnson, M. Clemens; Educational Uses of the Computer: An Introduction, Rand McNally & Company, Chicago, 1971

Kemeny, John G. and Kurtz, Thomas, E; Basic Programming, Second Edition, John Wiley & Sons, Inc., New York, 1971

Louis, F. B.; Contemporary School Mathematics - Computers 1 (with an introduction to binary arithmetic), Houghton Mifflin Co., Palo Alto, 1964

Martin, James and Norman, Adrian R. D.; The Computerized Society, Prentice-Hall Inc., Englewood Cliffs, N.J., 1970

Moursund, David G.; How Computers Do It, Wadsworth Publishing Co., Inc., Belmont, California, 1969

National Council of Teachers of Mathematics; Computer Oriented Mathematics/ an Introduction for Teachers, 1963

Organick, Elliott I.; and Stenberg, Warren; and Forsythe, Alexandra I.; and Keenan, Thomas A.; Computer Science: A Primer, John Wiley and Sons, Inc., New York, 1969

Rosenberg, Dr. Jerry M.; The Computer Prophets, The Macmillan Co., Collier-Macmillan LTD., London, 1969

Rothman, Stanley & Mosmann, Charles; Computers and Society, Science Research Associates, Inc., Palo Alto, 1972

Sack, John R. and Meadows, Judith L.: Entering BASIC, Science Research Associates, Inc., Palo Alto, 1973

Seligsohn, I. J.; Your Career in Computer Programming, Julian Messner, New York, 1967

Squire, Enid; The Computer: An Everyday Machine, Addison-Wesley (Canada) Ltd., Menlo Park, CA, 1972

The Math Group; Computer Conversations (Speaking in BASIC), Earl Orf and Diana Hestwood, Minneapolis, MN, 1973

Wirth, Niklaus; "Program Development by Stepwise Refinement," Communications of the ACM, April 1971, Volume 14, Number 4

COMPUTING ON A SHOESTRING

by David Dempster

Introduction This article is intended for secondary (grades 7-12) teachers and administrators who are interested in starting or expanding a program for the instructional use of computers. There are several key steps you must take to start a program using computers and a variety of possible ways to expand a program once you have established it.

There are several assumptions made regarding your present situation. First, you have no program or you have a minimal program with minimal facilities. Second, since the program is non-existent or new, there is a need for support (opinions and money) to make the program grow. Third, whoever is in charge of the program is knowledgeable about computers, computer programming, and the instructional use of computers. Fourth, the type of activities you want to do will require several volumes to present the topics.

There are a large number of factors you must take into consideration as you start to establish a program for instructional use of computers. Four of these factors which need immediate attention are program goals, types of computing facilities, courses (where in the curriculum do you start using the computer?), and authority.

Goals. The goals you establish will give direction for your entire operation. You need to have a clear idea of what you want the students to know and gain from the program. It is best to select one or two major goals which are pedagogically sound and realistic for your situation. Three examples of such goals are:

1. The students will be able to use the computer as a tool in problem solving by (a) constructing simple computer programs and/or (b) using computer programs developed by someone else.

2. The student will be aware of the importance of computers in our society and the impact computers will have on his life. (He will know what the computer can and cannot do, how the computer is used in our society today, what the potential developments will be, and he will learn not to have fear or be prejudiced about computers.)

3. The student will explore the career opportunities in the field of data processing and learn how the computer is used in business and industry.

Any one or two of these goals would be a sufficient base to build a program. There are a large variety of goals you could establish but the main point is that they must be established. Your individual preferences and values will determine your selection. However, make sure your goals can be achieved and are reasonable in terms of facilities, staff and time.

Facilities If you start with nothing you have several options for gaining computer facilities. The ideas discussed here are based on the assumption that you do not have much money available. For each option some of the advantages and disadvantages are mentioned.

First, someone could allow you to use their computer facilities free of charge. In most communities there are public and private computer facilities which are not used to their capacity. In some cases they may allow you to run student programs at no charge. Thus, one of the first things you do in investigating facilities is to study your community for available resources. The big advantage of this idea is that it won't cost you much. You may have to rent a communication device (usually a card punch device to prepare cards) at about $100 per month, but this is the minimum you will have to spend in most cases.

There are several disadvantages when using donated facilities. The programming language available often is not designed for student use. Programming languages such as BASIC are seldom available. Since many businesses operate a batch oriented system the type of activities you can involve students in is limited. For example, it is difficult to run many types of simulations or programs which require direct interaction with the computer. Since the facility has a major function other than education, your programs will be on a very low priority. This causes delays in having your programs run. In general, the turnaround time is longer than you may desire for educational purposes, but an innovative teacher can usually work around this handicap.

Second, if you are unable to find someone who will donate computer facilities, then look for someone who will rent you computer time on his system. Many computers sit idle after five o'clock or are not used fully during the day. This idea is similar to the first, but it has the added cost of rental fees. The cost will vary widely from place to place. The advantages and disadvantages are the same as mentioned previously.

Third, there are many organizations and companies which rent computer time on a large computer system on a timesharing basis. Normally you lease or purchase a teletypewriter and accessories which allow you to establish communications with the computer. Teletype rental fees vary from location to location, but one can usually be rented for about $80 per month plus installation fees. Computer time costs vary in rate and overall monthly charges will depend on how much time is used. Companies contacted can give estimates for rates. In addition you will need a telephone at about $14 per month plus an installation charge.

The advantages of this third option are that the teletype will be in the classroom, rapid turnaround time, student oriented languages (usually BASIC), a program library, and often storage capabilities. This is almost like having a large computer in your classroom, but not the responsibilities and problems which accompany such a system. The disadvantage is that you have only one (two if you're lucky) teletypes in the room. Trying to get one or two more classes of students time at the terminal is a major obstacle teachers must overcome.

The fourth option is to purchase a minicomputer for instructional purposes. There are several types of computer systems available for under $10,000. If the cost is spread over two or three years the yearly costs are much lower. Two possible configurations are (1) a minicomputer and a single teletypewriter for communication and (2) a minicomputer with a card reader which reads cards marked in pencil for communications. These two configurations are among the more common types currently used in schools. The first can be acquired for approximately $6,000.

The advantage of purchasing your own facilities is that you purchase a system which fits your needs. You can usually have the language, type of communications device, and amount of time you need for your program. Many of the small computer systems have the capacity for expansion as your program grows.

However, with your own facilities will come several potential problems. You may find that the small computer cannot handle large programs. The minimal system has no storage capacity such as magnetic tapes or disks. Also, you have the responsibility of keeping the system running. Since this is the most expensive and complicated method of acquiring facilities you need to do much planning and investigating before you make a decision. It will pay, in the long run, to get help from someone who knows the ins and outs of purchasing computer hardware and who can alert you to problems, both potential and real, which accompany such a purchase.

Courses. There are several ways a school can start using the computer for instruction. The three ideas presented here are the most common.

One method is to operate a separate course in "Computer Programming","Computer Science", or "Data Processing". This is a one semester or one year course designed to teach specifically about computers. Although there is as much rationale for offering this course as many others in the secondary curriculum, it is difficult for schools to add another course on top of already crowded schedules. The addition of this course may mean the elimination of some other course, which may or may not be desirable. In addition, a sufficient number of students will need to be recruited to actually operate the course.

Another method of implementing computer instruction is to offer a unit or section on or about computers as part of a regular course. The length of time is flexible (2 to 8 weeks) and depends on the instructor. This usually occurs in a science, mathematics or business course. This gives the instructor a large number of students and the opportunity to integrate the study of computers with the main course. The advantage of this approach is that it does not affect the school's schedule. When you attempt to add a new unit to an existing course something has to be changed. Either some topics in the course of study must be omitted or the depth of coverage must be changed to allow the new topic. This is no mean task for any instructor.

A third method of implementation is to teach an existing course using the computer. For example, mathematics could be taught using the compueer. There are several textbooks currently available which are structured in this way. Here the computer is a tool used in learning the subject under study. The normal subject matter is the focus of instruction, not the computer. Of course, the students must first learn to use the computer.

Authority. In every school system there are several people who have a definite say about what is done in the classroom. They are head teachers, principals, curriculum supervisors, superintendents, and school boards. Their opinions and goals directly influence what you actually do. If you are to make major changes in the curriculum you will need their approval even if you don't intend on spending much money. Once you enter the area of spending money the approval is more important and often harder to obtain.

Most schools have a method for teachers to propose innovation programs. The structure may be formal or informal, but it is there. Since you will have to work within the system you should investigate how it works in your school.

Most administrators are concerned with quality education and willing to help you develop a reasonable program. You need to gain their support if you want to promote your program. There are no surefire mehtods of winning the approval of administrators or school boards. By studying your system you are in the best position to judge what to do.

Ideally, a school would determine what it wanted to achieve (goals), develop an instructional program, and then acquire the appropriate facilities. There is a variety of reasons this approach is seldom used in implementing a program; you must make changes or concessions based on the realities of your situation. The four key points mentioned are not exclusive from one another. A change in one area causes changes in others. The goals, facilities, your overall curriculum structure, administrators and board, faculty, etc. all interact to determine your program. For example, if your facilities will be using a batch processing computer at some local business then you cannot expect to run simulations or teach the BASIC language. If the only teachers available are in the mathematics department then it would be difficult to operate a course in business data processing. These examples are just two of many types of situations which occur when you start planning.

Several alternative plans should be developed to take into account the various conditions with which you may finally have to contend. These plans should include decisions regarding the four points mentioned and other factors which exist in your situation. A tentative budget should be included in each plan.

When developing plans for a program you need to know specifically what you want now (or next year) and you need to have a long range plan for at least five years. This long range plan gives your immediate plans direction.

<u>School Board and Superintendent</u>. Before requesting facilities or money you need to convince people that the goals of your program are sound. The purpose of your presentation should be to convince the board or superintendent that the program is valuable. A general outline of such a presentation follows.

1. A major goal of education is to prepare students to function in our society. The computer is an integral part of our society. Alert them to the fact that they are in contact with computers every day. Many of the local businesses will be using computers. Point this out specifically, using companies they are familiar with.

2. Present the goals and rationale for each. For example, if the goal is: the student will be able to use the computer as a tool in problem solving, then the rationale is: the use of the computer in the secondary schools provides many advantages not otherwise available. A few of these advantages are:

   a. Generalizations, concept development, and procedures are often a major goal of instruction. The high speed and accuracy of the computer allows the emphasis to be placed on these major goals rather than just the answer.
   b. The use of the computer becomes a highly motivating device in the classroom.
   c. The use of the computer can eliminate the boredom of doing lengthy and complex computations. Also, it can eliminate errors made in computation. Thus, the student has accurate results to analyze and a

much longer time to spend on analysis.

d. The computer extends the student's ability to handle problems that were previously too difficult. Because of this, more realistic problems can be presented for study. Also, topics that were previously left out of the curriculum because of the computational complexities involved can now be included.

e. The computer can be used in every department in our secondary schools and can be used by every student.

f. Every college or university and many junior colleges and vocational schools have computing facilities available for students. Those students who will be involved in post high school training need to be prepared to make use of the existing facilities for their own advantage.

g. In addition to being able to use the computer for problem solving, the learning of computer programming is valuable because it teaches students to think logically. Computer programming is much more than just constructing a code for the computer. Programming involves several steps. They are:

   1. Definition of the problem.
   2. Analyze the problem and determine the most feasible method of solution.
   3. Design a solution - flow charting.
   4. Coding the solution in the programming language.
   5. Enter the program in the computer.
   6. Possible debugging.
   7. Run program.
   8. Possible debugging.
   9. Obtain print-out and analyze results.

   To be able to construct a program requires a high level of understanding of the concepts and procedures being studied. Everyone agrees that one must understand a topic very well to teach it to someone else. To teach a topic to a computer (programming) requires an even better understanding.

h. The use of the computer requires the active participation of the students.

i. "It is curious that in the 'normal' secondary school with its lessons to be memorized and exams to be passed, there is so little occasion for a student to ask his own questions and pursue their solutions with imagination and enthusiasm. If the solution to his problem is a working computer program, he will have learned as much as one learns a subject when one must teach it. In addition, he will possess a finished product that is very much his own creation." Digital Equipment Corporation

3. Put on a demonstration of your facilities if possible.
4. Leave sufficient time for questions and be prepared to respond adequately.

<u>Building a Program</u>. Once a program for instructional use of computers has been established it is often necessary to support it so that the program will be kept in the curriculum or so that the program can be expanded. The following ideas are activities which may help support a program.

1. Try to gain the support of other teachers in the building. You can demonstrate the computer in their classroom, run statistics on tests, or surveys, or help them on classroom research.
2. Gain support of school administrators by running statistical programs for them.

3. Present your program at a staff meeting. Keep it brief and positive. Stress what the students have learned and their attitudes toward the class.
4. Keep parents informed of what you are doing. Make a presentation to the PTA. Stress the "high quality" of the program and how modern it is.
5. Arrange for teacher inservice on educational use of computers. This can be done at a local college or at a local school. Most areas are able to provide instructors for such a course. (The school district will often pay for this type of inservice.)
6. Have the local newspaper do a short article about your program. This could include several pictures and a few short statements about the intent of the program.
7. Keep records of what the students have done (academically) on the computer, the amount of time per day the teletypewriter is actually in use, number of students involved in the program, opinions of students. At the end of the first year of operation a complete report of your progress should be submitted to all people concerned (superintendent, principal, department head, PTA, school board). This could contain the goals, student progress and achievement, (cost per student, optional) data that has been collected etc. It is the task of the teacher to be sure the program is successful. If for some reason it is not, find out why and support your reasons.
8. Several small drill and practice programs can be made available to simulate large computer assisted instruction (CAI) programs. These can be used in some mathematics courses to motivate students. This is time consuming but may prove valuable.
9. A brief overview could be prepared and distributed to other teachers, parents of students involved, newspaper, etc. The idea is to keep the concept alive in the minds of other people involved in education.
10. Once the program is underway it is helpful to keep the school board informed. A short 10-15 minute presentation is a good idea. Most school board members are interested in what is happening in the classroom. Find one or two students who would be willing to stand up and state what they feel they have gained from the course. A one or two page handout should be available to everyone at the meeting.

Conclusion. Whether you are starting or building a program for the instructional use of computers there are a tremendous number of decisions and plans to be made. To make your program successful keep in mind these four ideas. First enthusiastic teachers and students are necessary for any successful program. This gains support and gives your program momentum. Second, be prepared. You need to know where you are going and be able to support your proposals. Third, stay small and build on success. If you try to do too much at one time you are more likely to run into problems. Once you have failed to succeed it is harder to try a second time. Fourth, since your program is new you will need to prove it's merit. Nothing helps a program to grow more than success.

SECTION K: Abstracts of Related Articles for Chapter III

My Computer Likes Me*...*when I speak in BASIC
Dymax, Menlo Park, California

This short (63 pages!) delightfully written book is an excellent self instruction manual for learning the rudiments of BASIC on a time sharing system using a teletypewritter (TTY).

The theme throughout is, as written on the first page, "Experiment! Gamble! Guess,...then Try it!"

By varying a given problem the beginner is led through a step by step exploration of primitive BASIC. There are many examples and short exercises to help him. A typical problem: In year zero, we start with a population of P people. The population increases by R% each year. In N years, the population will be: $Q = P(1 + R/100)^N$.

Topics that are introduced are: line numbers, PRINT statements, LIST, RETURN, correcting mistakes, SCR, RUN, END, scientific notation, INPUT, GO TO, CNTRL C, ESC or ALT MODE, READ, DATA, Math Models, IF...THEN, INT(e), FOR-NEXT LOOP, STEP, SUBSCRIPTED VARIABLE, DIM, SUBROUTINES, double subscripts, information retrieval, etc. Strings and Files are omitted.

This is a clearly written book; a variety of type is used in printing to make each page eye catching. It is inexpensive ($1.19) and well worth the expenditure for the beginner.

Sue Waldman

*Recommendations for Academic Programs in Computer Science--A Report of the ACM Curriculum Committee on Computer Science
Communications of ACM, Vol. 11, No. 3 (March 1968)
Pages 151-197

This report can be described as a major contribution to the literature concerned with the development of educational programs in the computing and information sciences. It contains recommendations on academic programs at both the undergraduate and the graduate levels developed by the ACM Curriculum Committee on Computer Science. That portion of the report devoted to the undergraduate program is really a revised version of an earlier report entitled "AN Undergraduate Program in Computer Science--Preliminary Recommendations." (Comm. ACM 8,9 (Sept. 1965), 543-552; CR 7,1 (Jan.-Feb. 1966), Rev. 8767).

A classification of the subject areas contained in computer science is presented. These subject areas are grouped into three divisions: (1) Information Structures and Processes; (2) Information Processing Systems; (3) Methodologies; with several related subject areas, grouped into two divisions: (4) Mathematical Sciences; (5) Physical and Engineering Sciences.

It is somewhat distrubing to this reviewer to find numerical analysis (grouped under mathematical sciences) designated as a "related subject area".

There are twenty-two courses described in these areas, along with prerequisites, catalog descriptions, detailed outlines, and annotated bibliographies. Those readers interested in setting up academic programs will find this information extremely useful. They will also be quite pleased to find detailed recommendations for the undergraduate degree program.

Graduate programs in Computer Science are discussed, and some recommendations are presented for a master's degree program. However, doctoral programs are discussed only in very general terms with no specific recommendations. The importance of service courses in computer science and a minor in computer science are stressed, along with the importance of continuing education in this highly dynamic field.

Attention is given to the organization, staff requirements, computer resources, and other facilities needed to implement educational programs in computer science.

R. T. Gregory

*Reprint from the May 1968 issue of Computing Reviews

Conference Board of the Mathematical Sciences
Committee on Computer Education
Recommendations Regarding Computers in High School Education
2100 Pennsylvania Ave., NW, Suite 834, Washington, DC, 20037
April 1972

With computers rapidly coming into the educational scene, there is a pressure on secondary school teachers to teach something about computing. The character and quality of what is to be taught depends heavily on the guidance and materials available to these teachers. Seeing the need for national guidance, the CBMS issued this report after a 2½ year committee study of secondary computer education. They addressed the report to the mathematical-scientific community.

Their recommendations for preparation of courses and educational materials are as follows:

A. A "junior high school course in computer literacy designed to provide the student with enough information about the nature of the computer so that he can understand the roles which computers play in our society." This course would be required for all students and should be one semester in length.

B. An introduction to computing course, provided as a follow-up to the computer literacy course, with emphasis on problem solving. In addition, some modules integrating computing into secondary math courses and introducing simulation into the physical and social studies.

C. Special programs in computer science for gifted students.

D. A high quality, vocational computer training program.

Also included in their recommendations is National Science Foundation support for the development of a variety of programs for the training of teachers and the training of teachers of teachers of computer courses. It was suggested that a clearinghouse for information about high school computer education should be established.

The report does not go into detail concerning content of computer science courses or preparation of suitable educational materials but does give considerable direction for meeting their recommendations.

Jay Dee Smith

Allison, Ronald
Computer Programming for the Seventh Grade
Mathematics Teacher Vol. 66, No. 1, (January 1973)
Pages 17-19

Valley Stream North High School is a suburban school on Long Island, comprising grades 7-12. In September 1970, the mathematics department owned one Digital PDP-8L computer with 8K memory and four on-line terminals. The equipment was housed in a fairly large room called the Computer Lab. The room was used only for computer instruction.

During the eight-period day, two sections of computer mathematics were taught as a one-semester elective open to students in grades 11 and 12. For the remaining six periods, the lab was open for student use and was supervised by a mathematics teacher as one of his duty assignments. Students enrolled in the computer mathematics courses were permitted to use the equipment during the time when they were assigned to study hall or lunch. Since approximately forty students are enrolled during any one semester, two pertinent questions have arisen: 1) Are a capital outlay of approximately $15,000, a yearly service contract of $2,400, and eight teacher-periods a day economically justifiable for 80 youngsters? 2) Since the program is offered only to eleventh and twelfth graders and affects only a small number of mathematically oriented, college-bound students, is it sufficiently justified?

The decision was made to broaden the computer program to include as many youngsters as possible. Computer instruction was to begin in the seventh grade; further direction could be determined by the success or failure of that instruction.

A two-or three-week unit of computer instruction was given to all seventh graders. The unit included flowcharting, writing simple programs and hands-on experience. Each section was instructed in its regular mathematics classroom and was moved to the computer lab on a prescheduled basis for hands-on experience. When the instruction was completed, students were given access to the lab during their free time. High school students with experience in computer programming assisted in the lab.

As a result of this program, all 229 seventh-grade students were given at least a two-week unit of computer mathematics with a minimum of three days of hand-on experience. A survey of the students indicated that 227 of the 229 enjoyed their experiences and wished to continue such a program in the eight grade. One hundred and two students continued using the computer on their own time for both teacher-assigned problems and problems of their own choosing.

Some of the problems encountered included supervising thirty youngsters (with almost zero typing ability) trying to use four teletypes, preparing the teachers to teach the three-week units, and in coming up with meaningful problems that were capable of solution on the computer.

Rod Kohler

Amarel, Saul
Computer Science: A Conceptual Framework for Curriculum Planning
Communication of the ACM, Vol. 14, (June 1971), pp. 391-401

Two views of computer science are considered: a global view and a local view. The global view attempts to capture broad characteristics of the field and its relationships to other fields. The local view is concerned with the inner structures of the field: the kinds of knowledge, problems, and activities that exists within the discipline.

The Global View: Computer science is concerned with information processes and the classes of problems that give rise to them. It is therefore also concerned with general methods for solving problems with the help of information processing machines.

Activity in computer science seems to take two forms. One is concerned with system synthesis, exploration, and innovation: and the other oriented to analysis, search for fundamental principles, and formulation of theories.

The Local View: The formualtion of a solution or an algorithm is of primary concern in computer science. The language used to formulate the algorithm to be processed by a computer is also of major concern.

It is of interest in computer science to classify applications not by discipline, but by the kinds of problem solving methods, information structures, and procedures that are characteristic of the problems.

One important objective of computer science is to develop models and theories of computation that are responsive to the major design problems in the field.

Other concerns are in the areas of machine level languages, machine organization, and theory of formal languages. Results of the latter field are relevant to the design of high level languages, translators, and programming mechanisms.

Implications on curriculum planning are that there seem to be two broad areas of activity: (1) activities with problems, method of solution, and programming; and (2) activities with languages, schemes of processing, and design principles. Undergraduate study should provide a broad understanding of the two above areas and provide the skills needed to hold a professional position in the field.

Robert Layton

Austing, R. H. / Engle, G. L.
A Computer Science Course Program for Small Colleges
Communications of the ACM Vol. 16 No. 3
Pages 139-147

This report is authorized by the ACM Subcommittee on Small College Programs of the Committee on Curriculum in Computer Science. It gives recommendations of the content, implementation, and operation of a computer science program for small colleges.

Four basic courses are described, roughly corresponding to the areas of algorithms and programming (Course 1), applications of computers, (Course 2), machine and system organization (Course 3), and file and data organization (Course 4). The first course introduces the student to computer science and is a prerequisite for each of the other courses. The latter need not be sequential.

Implementation of the program requires student access to a computer with at least one higher-level programming language for student use. Cost can be justified only if computing services are used on a campus-wide basis. Hence, development of a community of computer science users on campus, as well as an excellent implementation of Course 1, is necessary. Minimal staff would be at least one full-time instructor, offering courses 1 and 2 each semester; and 3 and 4 each year. If possible, the school should hire a computer scientist to implement the program; if not, give a faculty member a year's leave to gain experience working in a computer science department. Some expansion of supplementary facilities, notably library holdings, should be provided.

Operation of the program is covered in detailed course descriptions which include:

Introduction: Outcomes, emphasis, what is included and not included, organization, and structure.
Catalog Description.
Outline: Main topics; % of class time for each.
Texts.

Brief recommendations for additional work are given, followed by an 88 item library list.

Sister Clare MacIssac

Bell, Frederick H.
Building a Conceptual Computer
Mathematics Teacher, v65, n1, (January 1972) pp57-60

Providing students with an introduction the the rapidly expanding field of computer science can be somewhat of a problem for the majority of teachers who do not have easy access to a computer facility. The seemingly inherent fascination that computers hold for many children can be subverted if a teacher's first lesson about computers does not provide proper motivation for students. Lengthy discussions of history, binary arithmetic, hardware, and programming can best be presented after the stage has been set by an initial informative and motivational lesson providing opportunity for student participation.

The following format for the first lesson about computers has been tested successfully with fourth graders, junior and senior high school students, college students, and teachers. In this approach, students are permitted to act out the functions of the computer components in a situation designed to conceptualize a computer. In the construction of our conceptual computer, the teacher, called the BOSS, will function as the control unit. A student, called the ADDER, will be selected to perform the arithmetic duties of the accumulator; a second student, the READER, will serve as the input unit; and a third student, the WRITER, will act as the output unit. Two additional students, a RUNNER and a LOOKER, are needed to respectively change data in storage and copy data from storage. Our storage unit will consist of two sections. Section 1, a large sheet of poster paper, will contain the list of instructions (the program) that the computer will follow in solving a specific problem. Section 2 of storage will be a set of small empty boxes, each one of which will be given an appropriate name corresponding to the name of the variable whose current value will be written on a card and stored in the box.

After assembling the computer components, we need a problem for the computer to solve. In selecting a problem, it is prudent to choose a relatively simple exercise in arithmetic so the computer will not be obscured by intricacies of the algorithm used in solving the problem. For motivational purposes it may also be desirable to find a problem that is new to the students. One problem that appears to satisfy both of these criteria is that of finding terms of the Fibonacci sequence 1,1,2,3,5,8,13,.... Students should be encouraged to construct the list of instructions for the computer to follow.

The computer simulated by people parallels the actions of a real computer. After students have been introduced to a computer in this informal manner, a detailed description of the operations of an actual computer can be discussed in relation to the conceptual computer.

Jon R. Lewis

Bernstein, Arthur
A Programmer Training Project
Communications of the ACM, Vol. 14, (July 1971), pp. 449-453

The purpose of the training project was to train black residents of the Albany-Schenectady area on computer programming. The project also arranged for jobs for the graduates. The course trained the students in FORTRAN on a GE time sharing system.

Jobs for the students were of primary concern for the people involved in the project. Managers were informed that the graduates of the program would not be competent programmers, but would be individuals that had demonstrated basic intelligence, ability to learn, and displayed an interest and gained some knowledge of computer programming. The intent was to provide later on-the-job training to bring the individual's skill up to the level of a computer programmer.

The only entrance requirement for the project was that the student must be black. No testing was performed, but selection was based on high school records if available. The ages of the students ranged from 20-35. Most of the students had poor mathematics backgrounds.

The class met two evenings per week for approximately seven months. One evening was devoted to two hours of lecture and the other evening consisted of open-ended tutoring. Homework was assigned each week. Complicated arithmetic expressions, double precision, and trigometric functions were omitted due to the students' poor math background. I/O and format statements were left to the end of the course.

The major problem in the class was going from the statement of the problem to an algorithm for its solution. The students also had problems using flow charts. About 20% of the entering students dropped within a few classes and another 20% left during the rest of the course. All 18 graduates were offered jobs. Six of these students became productive programmers. There was no correlation between high school performance and performance in the training program.

Robert Layton

Crowley, Thomas H.
APPLICATIONS OF COMPUTERS
UNDERSTANDING COMPUTERS, Chapter 11, McGraw Hill, pp. 101-111

This book was one of the very first "computer appreciation" books. It is available in paperback form at a reasonable price. A summary of a typical chapter follows.

If we look at the purposes of a computer, we can classify the applications of computers into three rather large basic categories. These categories are:

(1) To gain economy;
(2) To make the job feasible at all and;
(3) To achieve insight into some process.

For any practical use of a computer to be possible, at least one of these objectives must exist.

For example, computers are capable of storing large quantities of information and retrieving it rather rapidly by using magnetic tape or disc storage. Thus a company can save many dollars by committing payroll data (or sales reports, inventory, billing information, parts lists) to "memory". However, if it requires more work to put this information into a form acceptable to the computer than to do the original computation, the objective of economy is lost.

Computers are used in weather forecasting. Since it is possible to do a large number of complex calculations very rapidly and with very high accuracy, we now have reports, using data relayed from all over the world, ready within minutes of our data gathering. Therefore we now have information available in time to use it--not for records and statistical purposes only. Thus the computer may make a job feasible because of its speed and reliability of symbol processing. This is an example of a real-time application. Others are: controlling production lines in oil refining, cement manufacturing, electronic parts manufacturing and banking applications.

Simulations are used to gain insight into a given process or behavior of some model. We can, for example, simulate a given business over a given period of time and thus predict outcomes based on many different management policies. War "games" can be played. New machines or new processes can be "simulated" before being put into production.

Thus, as computers become faster and cheaper, only our imagination limits their use.

Gerald Larer

Dijkstra, E. W.
The Humble Programmer
Communications of the ACM, Vol. 15, No. 10 (October 1972)
Pages 859-866

E. W. Dijkstra was selected as the 1972 winner of the Turing award, given for outstanding contributions to the field of computer science. The Humble Programmer is the ACM Turing lecture given by Dijkstra. It is well written and very thought provoking.

Dijkstra traces the history of computers from a computer programmer's point of view, and his own involvement in this history. He notes that in the early days people tended to characterize programmers as being "puzzle-minded and very fond of clever tricks", and felt that programming was nothing more than optimizing the efficiency of the computational process.

Second generation computers were much more capable, and the needed level of software was much higher. Operating systems and compilers were good challanges to the computer programmer.

Dijkstra continues by recounting his feelings about the introduction of the third generation machines (IBM 360 series). "Then, in the mid sixties something terrible happened: the computers of the so-called third generation made their appearance."..."But the design embodied such serious flaws that I felt that with a single stroke the progress of computer science had been retarded by at least ten years; it was then that I had the blackest week in the whole of my professional life."

Dijkstra then goes on to discuss what is right and wrong about the current state of computer science. In particular he is critical of huge languages like PL/I, and is very supportive of "structured" programming languages. Many of his ideas are summarized by: "The sooner we can forget that FORTRAN ever existed, the better, for as a vehicle of thought it is no longer adequate: it wastes our brainpower, and it is too risky and therefore too expensive to use."

David Moursund

Dorf, Richard C.
Programming a Calculator
Introduction to Computers and Computer Science, Chapter 6
Boyd and Fraser Publishing Company, 1972

Programmable electronic calculators are constructed like digital computers. They rival digital computers for solving mathematical problems with small amounts of input data. Many have some of the features of a computer such as program storage, registers, floating point and fixed point numbers and arithmetic, logarithmic and exponential functions.

The line between a mini-computer and a programmable electronic calculator is not clear as many calculators have attachments similar to a mini-computer, such as a line printer.

The electronic calculator has the advantage of being portable and low-priced, and operates with reasonable speed.

Programming a calculator is somewhat similar to working with a computer. You usually need to flowchart, an algorithm to program a calculator, just as you do for a computer. Programming a calculator is much more difficult than a computer, because of the mathematical nature of the programming language.

Another disadvantage of the electronic calculator is that the operator must enter all the data into the calculator by hand.

John Stedman

Dorf, Richard C.
A History of Computers
Introduction to Computers and Computer Science, Chapter 2
Boyd and Fraser Publishing Company, 1972

Apparently man has always needed to process data and to calculate, and has invented various machines to help him. One early example, called Hero's odometer, adapted gears and wheels to count rotations. Another, the abacus, was used as early as 1000 BC. Stonehenge, an ancient British stone monument, illustrates a different type of caluclating device, possibly an astronomical observatory used to predict seasonal changes. Still another was the quipu, or knotted cord, the basis of an elaborate information system developed by the ruling Incas in the Peruvian Andes, some 500 years ago.

Coming to more modern times, we can name some of the men whose ideas and inventions are landmarks in the history of calculators and computers. Toward the end of the 16th century, John Napier developed logarithms and "Napier's bones", a simple device for multiplication. Others used his logarithm principle to develop the modern slide rule.

In the 17th century, Blaise Pascal invented the numerical adding machine and G. Wilhelm Von Leibniz made improvements still used in some of today's calculators. By the 18th century, progress in the clock-making industry and advances in metal-working led to greater mechanical precision. Early in the 20th century, mechanical calculators had become important tools in science and commerce.

One who contributed greatly to this development was Charles Babbage. By 1823, he had evolved several basic concepts of his "difference engine", designed to calculate with numbers, store information, select different ways of solving problems, and deliver printed solutions. His machine would have proceeded automatically once the instructions were fed into it by the operator. Babbage spent most of his life dealing unsuccessfully with engineering difficulties which prevented the realization of his dream: his machine was never built.

Meanwhile, however, other developments were paving the way for a later realization of all he had envisioned. In the early 19th century Joseph Jacquard developed a system of punched cards for automatically directing the intricate process of weaving. In 1890, Herman Hollerith adapted the punched card system to the special needs of census-taking. He also used electrical sensing and driving equipment.

The first modern digital computer, Mark I, developed by Howard Aiken and his associates, was presented to Harvard in 1944. Instructions and data were fed into it by punched paper tape. Its components were electric, electronic and mechanical. The first electronic digital computer was built by J. P. Eckert and J. W. Mauchly at the University of Pennsylvania in 1946. Analog computers were also developed about this time. These early computers were all designed to solve military problems. Not until 1954 with Univac I were computers commercially available.

In one sense, the modern electronic computer is a result of the evolution of earlier calculation machines under the influence of the new technology in radio, radar, and telephone transmissions. But technolgoy alone was not enough. Modern computers usually calculate in binary arithmetic, using a logical system invented by George Boole in the 19th century. Further, John Von Neumann, in the mid-40's, demonstrated how binary logic and arithmetic could work together in stored programs. In combination, all of these developments have made possible our modern data processing and computing facilities.

Sister Clare MacIsaac

Edwards, Judith L.
Computers.....a beginning
DCE Book developed under a NSF grant - Oregon State University - 1970 - 90 pgs.

Since virtually everyone is affected by computers in one way or another, everyone should understand their operation, capabilities, and limitations. We must find ways to bridge the "comprehension gap" that society has in the area of computeriz tion. The text is part of a program of instruction designed to place concise, basic, easily digestable information about computers into the hands of the person best qualified to disseminate it--the personnel of our public schools.

We need to examine the five basic functional units of a computer system and look at the organization of the Central Processing Unit. This should be done in very basic terms. Such topics as the different kinds of computers, the history of their development, and the binary number system should be discussed.

What kinds of problems are best suited for computer solution and how do they solve them? We need to look closely at the answer to this question and compare it to the different ways man solves problems. There are several important steps needed to prepare a problem for computer solution. Of these steps, the process of developing an algorithm and constructing a flow chart should be examined in detail.

How does man communicate with a computer? The text does not include any high level language programming but does take a brief look at different types of languages and includes a discussion of machine, compiler, and assembly language.

How is the computer used in education? It can be used to aid the administrator, the teacher, the student or the librarian. Administrative applications include such areas as accounting, scheduling, and student records. Instructional uses include computer managed instruction, computer assisted instruction, student problem solving, computer science classes, and simulations of real life situations. Other areas of application include the automated library and automated counselling systems.

Of course there needs to be some discussion on the social and cultural implications of the computer.

Robert Thomas

Franta, W. R.
Computer Arithmetic
Mathematics Teacher, v64, n5, (May 1971) pp409-414

Students new to computer programming often write computer programs that fail to perform as expected. Seeking to correct the resulting deficiencies, the student often looks in vain for an error in logic, when the true cause of the trouble may be failure to consider the nature of computer arithmetic. It is the purpose of this paper to point out the characteristics of computer arithmetic that may, if not considered when writing a computer program, cause certain programming statements to perform in a manner not intended.

To understand arithmetic as done by a computer, we must first know something about the way numbers are stored in a computer. Most computers store numbers in a format known as "floating point," which is much like that of scientific notation. We can easily see that only a finite set of numbers can be represented by the floating-point format. From another point of view, we may say that to store in the computer a number not having a floating-point representation, the number must be rounded off to the value of the nearest number that possesses a floating-point representation. Thus each floating-point number must represent all the real numbers in some interval.

The limitations of computer arithmetic as discussed can easily cause computer programs to produce strange and often seemingly impossible results if the effects of these limitations are not kept in mind.

Unfortunately, an attempt to organize the calculations of a program to minimize the propagation of round-off error is no guarantee that we have a computationally sound program. Many algorithms are sensitive to the approximations of numbers used in the computer and are thus unsuitable for computer use.

When FORTRAN programming was taught to college freshman over the past three years, it was noticied that students easily fall prey to pitfalls of the type described above. Since the nature of the pitfalls is a function of the arithmetic as done by the computer and is independent of the programming language used, it is to be hoped that slightly more emphasis by mathematics instructors on the above considerations would keep many students from falling prey to the pitfalls mentioned.

Jon R. Lewis

Horowitz, E.; Morgan, H.; Shaw, A.
<u>Computers and Society: A Proposed Course for Computer Scientists</u>
Communications of ACM, Vol. 15, (April 1972) pp 257-261

The purpose of the paper is to present a practical organization to make it easier to initiate interdisciplinary courses on the impact of computers on society. The main objectives of such a course are to educate computer scientists on the present and future impact of computers on society. The course is concerned with ethical questions and gives a humanistic perspective on the use and misuse of computers.

In a course of this type it is suggested that the first few lectures provide an overview of the present state of technology. This overview might include the following areas: 1) telecommunications; 2) manufacture of cheap large storage; 3) the growth of computer utilities; and 4) the proliferation of inexpensive minicomputers. The rest of the course should proceed from the devices to implications of the technology. The moral and ethical implications are left until the end of the course. The areas studied, relative to the impact computers have on them, are: 1) political, 2) economic, 3) cultural, 4) social, and 5) moral.

A course format is proposed which includes both lectures and discussion sections. Guest lecturers are encouraged. Students should be challenged to read, think, and to explore sensitive issues. Surveys, papers, and formal debates may be assigned. It is suggested that humanities and social science majors (with appropriate computer backgrounds) be mixed with computer science majors. Experience points to the need for a full-fledged "credit" course.

The course is designed to bring the perspectives of the sciences, social sciences, and the humanities to consider the question of the impact of computers on society.

Robert Layton

Kreitzberg, Charles B. and Schneiderman
The Elements of Programming Style
Journal of Data Education Vol. XII No. 5, February 1973
Pages 28-29

This article is an effort to apply a comprehensive set of rules for literary style, proposed by Professor William Strunk of Cornell University in 1919, to computer programming. There are many similarities between the writing of good prose and the writing of good programs. Both have rules of spelling and sentence formation. Variables correspond to nouns, operators to verbs, expressions resemble phrases, and statements resemble sentences.

Programmers seldom share a common background because of variations in compilers, operating systems, hardware, and programming environments. Each programmer must decide which rules will produce results and as in prose the best effect sometimes results from deliberate violations of rules. A good program is one which produces results: beyond this, some may be faster, require less storage, be more accurate, or structured to facilitate modificatic . In addition programs should, as much as possible, be compiler independent.

Professor Strunk's rules, and their applications to programming:

Work from a suitable design: Avoid the temptation to begin programming segments before planning the whole program. Programs should be modular, logical and meaningful so that debugging is easy.

Be clear. You are not likely to be the only person to read your program; and six months later even you will have forgotten what you've done. Insert comments, choose meaningful variable names and subprogram names. Provide good documentation.

Revise and re-write. Programs can usually be improved and usually are not modular. Therefore revisions should be considered.

Do not take shortcuts at the expense of clarity. They usually lead to trouble and seldom pay off. If you must shortcut, document your intent carefully.

Omit needless words. Make every statement count. Don't use unnecessary computations or unnecessary statements.

Prefer the standard to the offbeat. Avoid the great temptation to be cute and clever.

Do not use dialect. Try to avoid the niceties of a particular compiler. If the program becomes compiler-dependent, transfer to another system is difficult. Try to stick with ANSI standards where they exist.

Do not overwrite. There is a point at which programming effort leads to diminishing returns.

These rules are basic to good programming. Since taste and style are subjective every writer must make the final decision as to whether his program is suitable to his environment.

Frederic R. Daniels

LaFrance, Dr. Jacques and Roth, Prof. R. Waldo
Computer Science for Liberal Arts College
Wheaton College, Wheaton, Illinois
July 1972

A workshop, July 12-14, 1972, was held to discuss a computer science curriculum that would be relevant for the liberal arts college. In the past, curriculum proposals have been designed for universities. The eight members of the workshop team were mainly from smaller liberal arts colleges with an interest in computer science curriculum.

"There was agreement by those attending the workshop that studies in computer science definitely have a place in a liberal arts program irrespective of the institution's attitude toward vocational preparation."

Discussion at the workshop centered on four main areas:

1. The resources available and necessary for a computer science program. Most computers served a combination of academic and administrative functions. Budgets ranged from $24,000 to $40,000. The most common computer appeared to be the IBM 1130.

2. The obligations of the computer center to other departments and to students who need the requisite background for employment. A service course should be established enabling persons from all departments to learn how to make effective use of the computer in their discipline. There was varied opinion on the obligations to prepare students for employment in data processing.

3. A computer science curriculum was created.

4. The implementation of a computer science curriculum, beginning with a program that offers only an introductory course and moving to a full-fledged small college program. It was generally agreed that a small college would not be able to offer a computer science education of a caliber equivalent to that of a university.

"These recommendations represent the best thinking of the participants at the time of the writing of this report, but these recommendations are subject to ongoing evaluation and revision because of new information, experimentation, and dynamic change in the discipline."

Jay Dee Smith

Lewellen, Lee
COMPUTERS IN THE CLASSROOM
JOURNAL OF EDUCATIONAL DATA PROCESSING, v8, n? 3, (1971) pp. 33-38.

"Is the use of a computer a good instructional tool? How can the computers add strength to classroom instruction? Do students learn more, or faster, when a computer is used? In what subject areas do computers aid the teacher? Are schools in any number using computers in the classroom? How are the computers afforded?"

Traditionally, the educational system started late (that is, behind business, military, industry and science) in applying computer technology. And when they did begin, it was at the administrative level. At this time, however, the greatest growth is taking place in the use of computers in the instructional process.

The experiments which have been conducted thus far have shown that: "Students with varying degrees of academic proficiency invariably do better on College Board exams and Scholastic Achievement Tests after being exposed to computers in the classroom."

For example, Project Local, founded by five Massachusetts communities with help from the Title III E.S.G.A., declares: "Over the school year, the group which worked with the computer improved more than either of the other groups (one used flow charting--the other traditional) in general scholastic and reasoning abilities, as measured by standardized tests. ...Using the computer does seem to enhance learning, primarily by improving the student's problem solving skills and his understanding of the concepts underlying problem solution and it provides its own motivation force."

Even though not completely understood by researchers in the area, experimental work indicates that students are able to learn more in less time when the computer is properly used in the education process.

Included in this article are: (Information for lists below from survey conducted by American Institutes for Research)

1) Comparison of test scores for Project Local

Lists showing:

2) Computer applications for administration and instructional use
3) Extent of Administrative and Instructional Applications
4) Degree of Administrative and Instructional use among computer users

5) Nature and Purpose of Instructional use
6) Level and Source of Support of Instructional Use
7) Plans for Future use

Gerald Larer

Mosmann, Charles J.
Computers and the Liberal Education
Education Forum, Vol. 36, November 1971
Pages 85-91

Information science or computer science should be a required part of the curriculum of even the most humanistically oriented liberal arts schools. There are three major reasons for such an assertion. First, many people will be (are) directly associated with a computer in their professional life. This includes not just the programmer and systems analyst, but the people who tell programmers and analysts what they want the computer to do. Once, only engineers and scientists made use of computers, but more and more disciplines are finding the computer useful. Already, accountants, bankers, soldiers, architects, social scientists, public servants, doctors and lawyers are using computers; eventually all fields will be influenced. Therefore, everyone needs to know about the application of computers in his field: what is possible, what is difficult and what cannot be done (by a computer).

Second, many people will be involved with the computer in a passive way. Everyone is in contact with computers in their daily life; i.e., banking, billing, and mailing procedures are often computerized. The computer often stands between an individual and any large organization.

The third reason for including computer study in the liberal arts curriculum is that its implications in our society is in itself an appropriate topic of study and research. The age-old question of "What is man?" can be transformed into, "Can machines think? and if so, what makes man unique?" The impact computers are having and will have on individuals and society seems to be a futile area of research for those in the humanities and social sciences, as little research is being done here.

A school should be sure to include at least three areas of content in an introductory course in computers: 1) an introduction to computer technology and computer science. This would include what a computer can and cannot do, hardware and software explanations, how a computer works, etc. 2) An explanation of how problem solving with computers is done. This may include some rudimentary problem solving and programming. 3) A study of the implications of computers in our society now and in the future, with reference to ethical, economic and psychological issues.

David Dempster

Moursund, David G.
Selecting Goals for an Introductory Computer Programming Course
Will appear in Mathematics Teacher, November 1973 issue.

Computing is coming into the secondary school, generally through interested mathematics teachers. Computer science is a very large, expanding field that is still changing so rapidly that it has not yet stabilized at its most elementary levels. The teacher of a computer programming course will have to establish and defend the goals of such a course consistent with the facilities that he has to work with.

After the above background, Moursund discusses a variety of goals for an introductory computer programming course. These goals are independent of the particular teacher, student level, hardware, and software available.

Goal #1 To give the teacher training and experience in teaching such an introductory course. Most teachers lack formal training, and for (at least) the first several times through such a course, the teacher himself will be in a learning situation.

Goal #2 To increase a student's ability to solve problems, that is, a student must learn how to attach and solve new and different problems. A solution technique, algorithm, must be developed.

Goal #3 To familiarize the student with the types of situations in which a computer is applicable, that is, some of the capabilities and limitations of computers. He should learn to appreciate the problem-solving power of the man-machine combination.

Goal #4 To teach computer programming. This section is divided into three levels given as sub-goals: i.) to provide a student with the skills he needs for personal use in his areas of academic and vocational interests, ii.) to prepare the student to go on to a higher level of computer science (at the present, this is generally at a university level), iii.) to prepare a student to get a job in the computing field (generally, this goal is not attainable at the secondary school level).

Other goals are flow charting, debugging (detecting and correcting errors), and program segmentation (breaking the program into smaller units using subroutines).

The article is easy to understand and well organized. The goals given are of a general nature. For the teacher who wants to design a course, the article provides some desirable goals.

Jay Dee Smith

Rothman, Stanley and Mosmann, Charles
Computers and Society
Science Research Associates, Inc., 1972

The book, Computers and Society, appears to be suitable both as a textbook for a basic survey course in computer science as well as a readable book for the layman who wishes to investigate the roles of computers in our society. The authors have organized the book around three major questions--(1) Why study computers? (2) What are computers? and (3) What is the influence of computers on society?

In order to assess what computers will be able to do in the future, one must begin by separating facts from emotional arguments. Thus it is important for a person to have some idea of what a computer is and what it does. The reader is given some insight into the structure of a computer; however, the major portion of the book is given over to the influence of computers in our society. This is the question which concerns the majority of our citizens.

As with anything new and unfamiliar, many people are very concerned about the impact computers have on their lives. A considerable portion of the text is devoted to overcoming emotional reactions the reader may have toward computers. The authors do indicate that many concerns are valid but that numerous safeguards have been developed to insure privacy and prevent other forms of misuse. A good balance between the pros and cons of various uses of computers is maintained.

Computers are one of the major inventions of this century. They are becoming much more widespread as the number of terminals increases, as the cost of operation decreases, as the number of trained personnel increases, and as the number of applications increases. As the increase of computers continues it is necessary that more people acquire varying degrees of knowledge about computers and their application.

The major emphasis of the book was the coverage given the four major issues of concern in computer usage--automation and employment, privacy, individuality, and abuse of power. The coverage appeared well-balanced as each of the four areas was examined. Careful consideration of each issue by the reader should certainly be an aid in personal assessment of the computer movement.

A chapter is devoted to the future of mankind. Estimates are made as to when many technological improvements are likely to occur. In this examination of future happenings the authors see the computer as neither the savior nor as a destroyer of mankind. Three trends are noted in the book. These are an increased mobility of men, materials and information, future use of automation, and people's expectations. All point to increased use of computers in the future.

At the close of the book the authors have used a narrative format in presenting a Glossary of Computer Terminology. This seems like a good way to present vocabulary in the context of its usage. To locate specific terms within the novel format there is a Glossary Index.

This book appears to have two particular strengths--(1) it should be very readable by the average citizen and (2) a balanced viewpoint regarding computers is presented.

Ronald Little

Weizenbaum, Joseph
Computation Linguistics
Communications of the A.C.M., Vol. 9, No. 11 (January 1966)
Pages 36-43

ELIZA, a program developed at M.I.T., allows natural language conversation between man and computer. The program is written in MAD-SLIP for the IBM 7094 and operates on the MAC time sharing system at M.I.T.

ELIZA creates the impression that there is a human entity at a typewriter, somewhere, responding to each remark made by the user. This, of course, implies that the computer has a certain amount of intelligence. The purpose of this article is to explain the techniques used in creating this illusion and to dispel it.

Briefly, the program works like this. The user types in an input sentence. The computer analyzes the sentence using a set of decomposition rules, which, in turn are determined by key words appearing in the input sentence. Associated with the set of decomposition rules is a set of reassembly rules. It is this set of re-assembly rules that permits the computer to make responses to the user's statements.

ELIZA has many features which would allow it to simulate the responses of a psychotherapist or a counselor. Normal language conversation is the main feature of the system. This feature permits anyone to use the program, without having to learn a complicated language that the machine can understand. ELIZA also creates an illusion of "understanding." This ability allows the computer to appear to draw valid conclusions from the conversation. So important is this feature that the primary goal of ELIZA is to conceal its lack of understanding.

Mike Moser

Weizenbaum, Joseph
On the Impact of the Computer on Society
Science v176 (May 12, 1973) pp 609-614

Weizenbaum is a well known computer scientist, and has made substantial contributions to the field. This well written, thought provoking article, should be "must" reading for all people involved in the computer science field.

The computer has had very considerably less societal impact than mass media would lead us to believe. The Computer Industry is in a large part self serving. It is like an island economy in which the natives make a living by taking in each other's laundry. That which is not self serving is largely supported by the government and other industries that know the short-range utility of computers but have no idea of their ultimate social costs.

The impacts of the computer on society that will prove to be most important will be caused by the side effects of the computer. Is it reasonable to ask whether the computer will induce changes in man's image of himself and whether that influence will prove to be its most important effect on society? Weizenbaum thinks so, although he admits that the computer has not yet told us much about man and his nature. What is wrong is that we have permitted technological metaphors and techniques to so thoroughly pervade our thought processes that we have finally abdicated to technology the very duty to formulate questions for society.

Weizenbaum suggests that the computer revolution need not and ought not call man's dignity and autonomy into question. Is then the computer less threatening than we might have thought? Once we realize that our visions, possible nightmarish visions, determine the effect of our own creation on us and our society, their threat to us is surely diminished. But that is not to say that this realization alone wipes out all danger.

To this what are the responsibilities of the computer scientist? First, most of the harm computers can potentially entrain is much more a function of properties people attribute to computers than of what a computer can or cannot actually be made to do. Therefore the computer professional has an enormously important responsibility to be modest in his claims. Also the computer scientist must be aware constantly that his instruments are capable of having gigantic direct and indirect effects.

Larry Spencer

Wirth, Niklaus
Program Development by Stepwise Refinement
Communications of the ACM, Vol. 14, No. 4 (April 1971)
Pages 221-227

The ideas of structured programming and stepwise refinement are being pushed by some of the leaders of the computer science field such as E.W. Dijkstra, and Niklaus Wirth. This article details the ideas of stepwise refinement. Its introduction is quoted below. The article itself contains a long example illustrating Writh's ideas on stepwise refinement.

" Programming is usually taught by examples. Experience shows that the success of a programming course critically depends on the choice of these examples. Unfortunately, they are too often selected with the prime intent to demonstrate what a computer can do. Instead, a main criterion for selection should be their suitability to exhibit certain widely applicable techniques. Furthermore, examples of programs are commonly presented as finished "products" followed by explanations of their purpose and their linguistic details. But active programming consists of the design of new programs, rather than contemplation of old programs. As a consequence of these teaching methods, the student obtains the impression that programming consists mainly of mastering a language (with all the peculiarities and intricacies so abundant in modern PL's) and relying on one's intuition to somehow transform ideas into finished programs. Clearly, programming courses should teach methods of design and construction, and the selected examples should be such that a gradual development can be nicely demonstrated.

This paper deals with a single example chosen with these two purposes in mind. Some well known techniques are briefly demonstrated and motivated (strategy of preselection, stepwise construction of trial solutions, introduction of auxiliary data, recursion), and the program is gradually developed in a sequence of refinement steps.

In each step, one or several instructions of the given program are decomposed into more detailed instructions. This successive decomposition or refinement of specifications terminates when all instructions are expressed in terms of an underlying computer or programming language and must therefore be guided by the facilities available on that computer or language. The result of the execution of a program is expressed in terms of data, and it may be necessary to introduc further data for communication between the obtained subtasks or instructions. As tasks are refined, so the data may have to be refined, decomposed, or structured, and it is natural to refine program and data specifications in parallel.

Every refinement step implies some design decisions. It is important that these decisions be made explicit, and that the programmer be aware of the underlying criteria and of the existence of alternative solutions. The possible solutions to a given problem emerge as the leaves of a tree, each node representing a point of deliberation and decision. Subtrees may be considered as <u>families of solutions</u> with certain common characteristics and structures. The notion of such a tree may be particularly halpful in the situation of changing purpose and environment to which a program may sometimes have to be adapted.

A guideline in the process of stepwise refinement should be the principle to decompose decisions as much as possible, to untangle aspects which are only seemingly interdependent, and to defer those decisions which concern details of representation as long as possible. This will result in programs which are easier to adapt to different environments (languages and computers) where different representations may be required.

The chosen sample problem is formulated at the beginning of section 3. The reader is strongly urged to try to find a solution by himself before embarking on the paper which--of course--presents only one of many problem solutions."

DAVID MOURSUND

Zobrist, Albert and Carlson, Frederic
<u>An Advice-Taking Chess Computer</u>
Scientific American, (June 1973),Vol. 228, No. 6
Page 92 - 105

The field of artificial intelligence (can a computer be programmed to think?) is one of the more interesting and challenging aspects of computer science. The problem of writing a computer program to play a good game of chess has a long and progressive history. An annual computerized chess tournament has helped to stimulate renewed interest in this particular problem.

The authors of this article work at the University of Southern California. They have written a chess playing program which they feel is fundamentally different from ones written previously. It has a built-in ability to "take advice" from a person. The author's program was defeated in the most recent U.S. computerized chess tournament; however, the authors feel the poogram will continue to improve as it receives appropriate advice from their consultant (a senior chess master).

This is an excellent introductory article for a student who has some background and interest in chess. It traces the development of many of the key ideas and pinpoints the current major problems. The article ends with a listing and discussion of a game between the computer and a person.

David DuBose

CHAPTER IV

COMPUTERS AS AN AID TO LEARNING

by

Dick Bach*

Fred Board

Cliff Burns

Lloyd Fraser

David Moursund

Jan Moursund

Tom Stone

Wally Waldman

*Dick Bach of North Eugene High School, Eugene,Oregon, served as a group leader for this chapter.

SECTION A: OVERVIEW by David Moursund

Introduction In this Handbook we have divided the topic of instructional use of computers into three major subtopics. *Chapter III discusses teaching about computers,* including teaching computer programming. Chapter IV (this chapter) is concerned with use of the computer as an aid to learning. Some people call this topic "computer assisted learning" to distinguish it from "computer assisted instruction" covered in Chapter V. In computer assisted instruction the computer serves as a delivery system--it presents instruction. The dividing lines are not clear cut, and it is certainly not critical that one should classify each instructional use of computers into exactly one of the three general categories.

The field of computer assisted learning (CAL) can itself be divided into two subfields. Many aspects of CAL do not require knowledge of computer programming either on the part of the teacher or on the part of the student. *Both the teacher and the students use the computer, but* they use programs which have been written by other people and stored in a computer library. (Occasionally the program writer will be the teacher or a student in his class). Sections B, C, and D of this chapter are devoted to a discussion of various aspects of this form of CAL.

A second major subdivision of CAL is concerned with the situations in which students are expected to write computer programs as an aid to their studies of some field such as mathematics, business or science. Generally a high level of programming skill is not necessary. However, both the teacher and the student will need some programming skills. In either case the person with very minimal or inadequate skills will be handicapped.

*The remainder of this section is devoted to a brief dis*cussion of the topics discussed in more detail in subsequent sections in this chapter.

Simulation One of the most fundamental concepts of science is that of prediction--that is, of constructing models which accurately describe some aspects of the real world. A formula such as $d = .5at^2$ relating distance to acceleration and time is an example of a model. It can be used to predict the position of a falling object (acceleration due to gravity) after a certain time period. The model assumes that the object moves in a vacuum under a constant acceleration; neither of these assumptions may be correct in a typical application of the model. That is, a model need not be perfect to be useful.

The concept of modeling can be applied in the social sciences, business, government, military, etc. The word "simulation" is often used in these more general contexts. Moreover, the complexity of the models often makes a computer helpful, or even necessary, in actually using the model. A model (simulation) of the economy of the United States, for example, might involve several hundred variables. To use the simulation to predict the effects of a tax cut might require billions of computations.

Packaged Programs Many thousands of computer programs have been written to solve problems--math problems, engineering problems, statistics problems, etc. Often it is possible to write a computer program with sufficient generality so that it can solve all of a certain kind or type of problem. Thus a program can be written to compute the correlation of two sets of n numbers. It makes no difference whether the actual data to be analyzed are sets of student test scores, or sun spot counts vs. stock market activity.

Over the years many hundreds of general problem-oriented programs have been written, carefully tested and documented and published. The typical computer center maintains a library of such "packaged" or "canned' programs. Packaged programs of this sort can be an object of study in their right, or can be used in studying the subject matter which happens to require a particular computation. In either case they play an important role in a modern educational system.

Information Retrieval A computer is a machine designed for the input, storage, manupulation, and output of information. If one downplays the emphasis upon manipulation (i.e., especially arithmetic computations) then what is left is the concept that a computer is a machine designed for information retrieval.

In the world of education, computerized information retrieval has made significant progress in two areas. In school administration a computer can be used to store student information (names, address, parent information, course schedules, academic record, etc.) in a form to allow easy access. This is widely done. In educational research the computer can be used to store huge amounts of information (for example, titles and short abstracts of all journal articles published in a certain field) in a form to allow rapid, comprehensive access.

A third major aspect of computerized information retrieval is largely not yet realized in public education. Eventually we can expect that the previously mentioned tool being made available to select researchers will grow in power and be made available to all students. One can think of this as the computerization of libraries. A student will be able to "look up" information on a topic while sitting at a computer terminal. He will have access to a huge central library independently of the size or remoteness of the particular school he is attending. This library will be much more comprehensive than could be afforded by any single school.

Computer Programming As An Aid to Learning Almost all students can learn to program a computer at a level which is useful to themselves and useful in studying subject matter such as mathematics, business or science. Of course, learning to program a computer is a non-trivial task, and what constitutes a useful level of skill is subject to debate. At the college level many fields of study (i.e. business, engineering) require all of their students to take at least one course in computer programming. One might conclude that the material covered in one college level introductory programming course is a useful level.

Computer programming can be taught at the grade school level. It is fairly common to find students learning to program at the junior high school level. As an example, Mike Neill at Roosevelt Junior High School in Eugene, Oregon offered a nine weeks introductory programming course each quarter for several years (1969-71). The course was open to the general "run of the mill" seventh and eighth graders.

It is possible, then, to teach a significant amount of computer programming to a typical 7th or 8th grade mathematics class student. If this is done, then many of the mathematics topics at these grade levels can be examined in light of this new, additional tool.

If this can be done at the junior high school level, then clearly it can also be done at the senior high school level. The National Science Foundation supported the "Colorado Project" (1) which developed a second year algebra and trig course involving the use of computers. This course is now offered at a number of high schools. In Oregon a Colorado Project "workshop" was held in summer 1972 and again in summer 1973. Thus, quite a few teachers in Oregon have studied these materials fairly carefully and are involved in using them in their high schools.

Quite a bit of instructional materials involving students writing programs have been written for use at the college level. Perhaps the best known materials are the CRICTSAM (2) materials--an introductory calculus course using computers. Although these materials have received wide national attention, they have not been used much in Oregon. An exception is Lane Community College in Eugene, which has offered such a course several times.

Conclusion The use of computers as an aid to learning is just beginning. As we make further progress in computer hardware, software, and instructional materials, it will become commonplace. Eventually students will make everyday use of computers in the manners discussed in the following sections.

References

1. Beavers, Collins, Herber, and Larsen: Second Course In Algebra and Trigonometry With Computer Programming; The University of Colorado,1973.

2. Calculus: A Computer Oriented Presentation; Florida State University, 1968.

3. Johnson, David: (Editor and author of several parts): Computer Assisted Mathematics Program (6 Volumes). Scott, Foresman and Company, 1969.

SECTION B: SIMULATION

Introduction

Imagine yourself as a visitor to City High School. As you walk down the hall you pass a small room bustling with activity. Upon entering the room you observe four enthusiastic students surrounding one of three teletype terminals in the room. You decide to take a closer look to see what has them so excited. You find that they are engrossed in making football strategy decisions. They are using the terminal to interact with a computer program that has been written to simulate a football game. The students have the ball on the computer's 30 yard line with a third down and 12 yards to go situation. The students have decided upon their strategy and the student seated at the keyboard types in the number that is the code for that strategy. The students moan upon seeing the following message: "Intercepted pass. Return to 50 yard line. Select your defense." Now you move on to the next terminal at which a girl is interacting with a 3-dimensional tic-tac-toe program called QUBIC. This game requires her to visualize, in three dimensions, the two-dimensional display before her. A cheer from the girl seated at the third terminal interrupts your observation of the QUBIC game, and you walk over to the third terminal. The girl relates that she has just "made a killing' on the stock market in a simulated stock market game.

While placing yourself in the role of an observer in the above paragraph, you have actually experienced a form of simulation. Simulation can be thought of as a model for a given situation. Usually it attempts to replicate the essential aspects of reality, so that reality may be better understood or controlled. "Gaming" is a term often associated with simulation. It may be described as a procedure involving competition and requiring decision-making to achieve pre-specified goals. There is not complete agreement within the field about the classification of simulations and games. Some experts believe that games are a type of simulation; some feel that certain games are simulations while other games are in a classification of their own; and still others feel that games and simulations are separate entities. There is, however, agreement that games are competitive by nature while simulations are not necessarily competitive. Also, simulations usually attempt to produce a realistic situation while concern forrealism in games is not usually necessary. To point out these differences, consider two examples. First, the testing of a rocket nose cone in a wind tunnel which attempts to produce conditions similar to those it would encounter upon entering the earth's atmosphere is a simultion. Note that competition is not a factor. Second, the popular game "Monopoly" has competition as its main emphasis even to the point of distorting reality. A goal of simulation is the transfer of experience so that the learner is better prepared to cope with real events, while this is not necessarily a goal in gaming. Further mention of gaming will be made later in connection with the computer's role in education.

General Use of Simulation

"The field of simulation/gaming has passed through three phases since the 1950's: Acceptance on faith through 1962 or 1963, a 'Post-Honeymoon' period during the years 1964 to 1965 when the first crude attempts to evaluate simulations and games led to disenchantment, and the present period of 'Realistic Optimism' based upon accumulated experience and further experimentation."[1] A teacher who has never used simulation or gaming for instructional purposes must decide if they would be valuable learning processes, and if so, how they could be incorporated into his discipline. There are two ways that one could use simulation and gaming in his teaching. First, the modeling process could be studied; this would involve learning how to write and analyze simulations and games. Second, simulations and games could be used as a means to learn about something else. This section is concerned with the latter idea, and will now proceed to list some of the possible advantages and disadvantages of using simulations in this manner.

Advantages of simulation:

1. Emphasizes the enquiry approach to learning.
2. Provides experience in things too expensive or dangerous to be otherwise feasible or practical.
3. Permits expansion or compression of real time.
4. Forces students to take an active tole in learning.
5. Increases self-motivation.
6. Promotes discussion because of common student experience.
7. Is expected to improve retention of transferable action (but evidence is not yet available).
8. Eliminates some measurement difficulties.
9. Requires clarity of communication (both in following instructions and in expressing ideas).

Disadvantages of simulation:

1. May require increased teacher planning time.
2. May require too much class time to complete.
3. Results in lack of teacher control over what is learned.
4. Is often expensive.
5. May create a classroom situation in which students are talking and moving about the room.
6. Is difficult to validate as an effective learning instrument-
7. May produce conformity in students.

As is true of just about any educational instrument, there are times when simulation is highly effective and other times when it is not. The advantages listed above suggest that it can be done effectively when a student would benefit from a real-life

[1] Boocock, Sarane S. & Schild, E.O, Simulation, Games and Learning, Sage Publications Inc., Beverly Hills, California.

experience, but for reason of time, cost, danger, etc., the true experience would be inpractical or impossible. However, before using a simulation, the advantages and disadvantages need to be weighed and considered with the context of course work and the teacher's willingness to plan and experiment before it can be effectively utilized.

Computerized Simulation

The preceding paragraph dealt very briefly with the general use of simulation in education. However, since this Handbook is devoted to the use of computers in education, the remainder of this section will deal with computerized simulation.

Why should a computer be used in simulation? An obvious answer is that many simulations require an immense amount of computation. There are a number of other reasons. A teacher may be reluctant to use simulation or gaming in a course because of the amount of time and effort required for teacher preparation. The preparation time can be greatly reduced if the teacher is able to gain access to a computer system and locate a prepared computer program of the desired simulation. (Location of prepared simulations will be discussed later.) Also the in-class student time required to complete a simulation can often be reduced significantly for two reasons. First of all, single students or small student teams may conduct the simulation at a computer terminal while the rest of the class works on something else (or the simulation could even be done outside of class time). Secondly, many educational simulations and games that are not computerized require interaction between the teacher and students or between student teams, and responses are required at both ends of the interaction, which can be quite time consuming. The computerized simulation can speed up this process by providing immediate responses at the computer end of the interaction. As is true for simulation in general, computer-aided simulations provide learning experiences which might not be available to students because of factors such as safety, equipment cost or availability, prohibitive time, or other factors of cost of convenience. Simulations may also provide to students an instruction approach which will enable them to attain a lesson's objectives with greater speed and/or ease. An example of this would be an acid-base titration simulation experiment used in a chemistry class. Because of technological laboratory developments, a chemist rarely needs to perform a titration manually. Recording PH Meters and automatic Burettes have made typical high school laboratory titration procedure obsolete. Therefore, the acid-base titration is important, not as a laboratory technique, but as a means for demonstrating the application of chemical principles. The student may now use a computer simulation to demonstrate the application of these concepts in a way that might be faster, safer, and cheaper than performing the laboratory experiment. Some other advantages that computer aided simulations provide are (1) instantaneous feedback to the student who has just made a decision in a simulation, (2) quickness in performing many rigorous and/or lengthy computations, (3) easy access, and the use of models of a greater complexity.

How can a teacher with little or no computer experience begin using computerized simulations? (Here we are assuming that time-shared computing facilities are available in the school. Of course, computer simulations can also be run on a batch processing computer facility.) Such a teacher may be apprehensive, but in reality it can be very easy to use a computer-aided simulation. If a teacher is interested in trying a simulation, the first thing he should do is attempt to find a teacher who has had experience in the area. There is generally at least one teacher in a school district involved in some form of computer-aided instruction. In larger districts there is probably a teacher in each high school that could be used as a resource person. This person could explain the hardware and software facilities available and demonstrate their use. A typical case would involve going into a classroom where a teletype terminal is located and typing several short instructions on the keyboard just as would be done on an electric typewriter. This would initiate a response from the computer. At this point the person at the teletype would have the option of calling into use one of the prepared simulations or other programs available in that system's library. He could then continue interaction with the computer until the program was completed. In this way the novice could be introduced to the use of a computer under relaxed conditions, thus relieving the original apprehension.

There are available a large number of instructionally oriented simulations that represent many disciplines and which use the computer to varying degrees. Most of these are written to be used with the interactive process because of the advantage of immediate response. However, there are some simulations that lend themselves very well to batch processing.

Once a teacher is aware of the facilities available to him and is somewhat familiar with their use, he must then initiate his students to the use of computerized simulations. Just like teachers, many students are also apprehensive about their initial encounter with a computer. This would be an excellent time to use a computerized game, like the game of football mentioned earlier, which would provide the means of a painless introduction. Because of the nature of gaming, students can easily become addicted to it. Thus the teacher needs to take great care so that after the introduction to the computer, any simulations or games performed should be used only as the means of accomplishing an educational goal. For example, assume an economics class is dealing with the stock market. A simulation dealing with the daily or weekly transactions of buying, selling, and trading could be an excellent means of giving the students a feeling for how the stock market operates. But the simulation needs to be preceded by discussion of the stock market, a study of necessary economic concepts, and other information needed to enable the students to activate the terminal and perform the simulation. Follow-up activities similar to those that would be used after seeing a movie on the stock market should also be planned. In other words, a simulation is just a tool used to help a student gain understanding of his environment.

How can a teacher locate prepared computerized simulations? The best source is probably a catalog listing and description of all prepared programs, including simulations and games, that are available from the system's library of stored programs. Depending on the system being used, the list of simulations could be quite extensive. For example, the Oregon Total Information System (OTIS) Hewlett Packard User's Library of scored programs contains over fifty simulations and games. Not only are library programs easily located, but they are also quickly and easily accessed. For additional sources, one should refer to the source bibliography at the end of this article. It is also a good idea to examine subject area journals such as Science Teacher, Social Educator, Journal of Chemistry Education, Mathematics Teacher, and American Journal of Physics. Once a teacher goes beyond his computer system's library of stored programs, however, computerized simulation becomes more difficult; this brings up the question, how can a teacher modify a prepared simulation to meet his particular needs? There are two reasons why one would need to make modifications. First, the program may need to be altered in order to make it compatible with the computer system being used. This could range from a simple change in format to a possibly complex translation into a different programming language. The other reason for altering a program would be to change the actual simulation so that it would be a more effective learning aide. This too could require either a minor or very complex modification. In either case the novice programmer will no doubt want to seek help. On the other hand, the teacher with some programming experience may decide to make the changes himself, depending on the degree of complexity of the modification and the extent of his programming ability.

Computer-aided simulations are, of course, developed by people. There is no theoretical reason why a teacher might not develop his own simulations. In actuality, however, it is generally a time-consuming and difficult task to develop a good simulation. Thus, it is a task that should not be lightly undertaken by the typical (overworked) teacher.

EXAMPLES

The following are four examples of simulations. These were taken from the Hewlett-Packard Library. (These are representative of the approximately seventy simulations listed in the H.P. Manuals.)

The first example is called POLUT. It was developed by the Huntington II Project. An actual run is shown in order to illustrate (1) the format, (2) the variety, (3) the input needed by the students, (4) the preparations needed by the students, (5) the different output displays, (6) the accompanying instructions in the program, and (7) other ...res (such as language, error messages, etc.) Comments are written throughout the output from the computer run. Note that the underlined portions are the responses of a student (i.e., student input).

```
                    WATER POLLUTION STUDY

INSTRUCTIONS (1=YES, 0=NO) ? YES --------------(Note that student
                                         makes a mistake here.)
? INPUT DATA NOT IN CORRECT FORM--PLEASE RETYPE

                    (Machine is programmed to respond to such errors
?1
IN THIS STUDY YOU CAN SPECIFY THE FOLLOWING CHARACTERISTICS:

A.   THE KIND OF BODY OF WATER:
     1.   LARGE POND
     2.   LARGE LAKE
     3.   SLOW-MOVING RIVER
     4.   FAST-MOVING RIVER

B.   THE WATER TEMPERATURE IN DEGREES FAHRENHEIT:

C.   THE KIND OF WASTE DUMPED INTO THE WATER:
     1.   INDUSTRIAL
     2.   SEWAGE

D.   THE RATE OF DUMPING OF WASTE, IN PARTS PER MILLION (PPM)/DAY.

E.   THE TYPE OF TREATMENT OF THE WASTE:
     0.   NONE
     1.   PRIMARY (SEDIMENTATION OR PASSAGE THROUGH FINE SCREENS
          TO REMOVE GROSS SOLIDS)
     2.   SECONDARY (SAND FILTERS OR THE ACTIVATED SLUDGE METHOD
          TO REMOVE DISSOLVED AND COLLOIDAL ORGANIC MATTER)
***************************

BODY OF WATER ? 3
WATER TEMPERATURE?  64          (Input underlined)
KIND  OF WASTE? 1
DUMPING RATE? 114

NEW YORK CITY ONLY POLLUTES ITS WATER AT THE RATE OF 12 PPM/DAY.
MAKE YOUR RATE BETWEEN 0 AND 14.  ---------- (Note the response
                                         and comparison.)
DUMPING RATE? 6
TYPE OF TREATMENT? 1

DO YOU WANT:  A GRAPH (1), A TABLE (2), OR BOTH (3) ? 3
```

TIME DAYS	OXYGEN CONTENT PPM	WASTE CONTENT PPM
0	8.11	2.67
1	7.9	5.35
2	7.54	7.43
3	7.23	9.05
4	6.98	10.31
5	6.79	11.28
6	6.64	12.04
7	6.52	12.63
8	6.43	13.08
9	6.36	13.44
10	6.3	13.71
11	6.26	13.93
12	6.23	14.09
13	6.2	14.22
14	6.18	14.32
15	6.16	14.4
16	6.15	14.46
17	6.14	14.5
18	6.14	14.54
19	6.13	14.57
20	6.13	14.59
21	6.12	14.61
22	6.12	14.62
23	6.12	14.63
24	6.12	14.64
25	6.12	14.65
26	6.11	14.65
27	6.11	14.65
28	6.11	14.66
29	6.11	14.66
30	6.11	14.66

```
     0...OXYGEN-SCALE.....5...OXYGEN-SCALE...10...OXYGEN-SCALE...15
     0..WASTE.10..SCALE.20..WASTE.30..SCALE.40..WASTE.50..SCALE.60
DAY  I---------I---------I---------I---------I---------I---------I
 0   I W                             O
 1   I  W                            O
 2   I   W                         O
 3   I    W                       O
 4   I     W                     O
 5   I      W                   O
 6   I       W                  O
 7   I        W                O
 8   I        W                O
 9   I        W               O
10   I         W              O
11   I         W              O
12   I         W              O
13   I         W              O
14   I         W              O
```

WASTE CONTENT AND OXYGEN CONTENT WILL REMAIN AT THESE LEVELS UNTIL ONE OF THE VARIABLES CHANGES.

ANOTHER RUN (1=YES, 0=NO) ? 1

Notice the outline structure that the simulation is written in, thus making it fairly easy to follow. If an individual were a slow reader, he is not rushed to finish the reading. After the teletype has printed the words "BODY OF WATER?", it will do nothing else until the student responds by typing in a number from "1" through "4". The student is then asked to prescribe the temperature of the water and the specific type of waste. Then he is allowed to determine the amount of waste placed in the water and the specific type of treatment. (Again this is done by typing numbers which act as codes.) After typing all of the information, the student is even given a choice of displays (graph or table or both). However, this does not end the simulation as the student gets a chance to redo the simulation and thus change any parameters if he wishes. In making repeated runs with some responses changed, the student could obtain comparative data and thus try to determine some conclusions as to (1) the effectiveness of the types of treatments, (2) the difference in the ease of polluting various types of water, (3) the effect the temperature has on the pollution of a body of water, etc. The instructor could determine if a student could interpret tabular or graph data.

In the next three examples a description of the programs will be provided, but no actual runs will be produced as was done for the first example.

LABOR consists of an interaction, representing collective bargaining, between students. One or more groups of students are used for several periods. (The time is optional.) Each group splits into two smaller groups to represent the role of management and the role of labor. The simulation involves the following issues: (1) duration of contract, (2) wage rate, (3) work hours per week, (4)overtime and overtime wage rate, (5) vacation days per year, (6) number of paid holidays, (7) days of sick leave, and (8) bonuses and insurance. Before using the simulation, the student should be familiar with each of these issues. During the bargaining, if no agreement is reached in a length of time, pressure is applied to both management and labor to end the strike. If a deadlock occurs in negotiations or the negotiations extend for a "long" time, the computer arbitrates and both sides must agree to the terms. More interaction before and during the simulation is accomplished between students in the LABOR example then in the POLUT example. (Indeed, this simulation can be run without the use of a computer if desired.)

The third simulation is entitled WHEELS. This model simulates some of the economics related to the ownership of an automobile. It involves the purchase and maintenance of a car for a period of one year. Some of the facets involved are (1) the method of financing, (2) the choice of insurance, (3) the method of recording the expenses, etc. The computer plays a very active part in this simulation in that it randomly determines daily expenses, unexpected events such as accidents, major repairs, and increase or decrease in insurance rates. This program is dependent on other materials that can be purchased for $2.50 for a sample set for a teacher or $40.00 for enough to outfit a class of 35 students. (All of the materials may be used repeatedly, thus making the initial cost the only one.)

The fourth example is called BOHR. The computer graphically displays hydrogen emission spectra and the student must decide which energy level transitions are responsible for the lines of the spectrum that he has chosen. If that is done accurately, the student is then asked to determine the energy of photons emitted by the electron as it 'falls' between certain energy levels after some explanation is output by the computer. However, if the student did not estimate the energy level transitions correctly, then he is tutored by the computer through six steps of instruction. This simulation is independent of supplementary material such as was needed for example three.

There are some similarities common to all four of the simulations described. However, there are also many differences. This is true of all computer-aided simulations. Thus, it is imperative that an instructor be fully aware of the content, nature, and structure of any computer-aided simulation that he wishes to utilize in a teaching situation.

BIBLIOGRAPHY

Publications and Organizations

Computer Explored Series
Control Data Institute
Division of Control Data Corporation, 8100 34th Avenue South
Minneapolis, Minnesota 55420

Computing and Mathematics Curriculum Project (CMCP)
Division of Mathematical Sciences
University of Denver
Denver, Colorado 80210

DEC Program Library
Digital Equipment Corporation
Maynard, Massachusetts 01754
(P,S,G)

Games Central
55 Wheeler Street
Cambridge, Massachusetts 02138

The Guide to Simulations/Games for Education and Training
Second Edition. David W. Zuckerman and Robert Horn
Information Resources
P.O. Box 417
Lexington, Massachusetts 02173

H.P. Basic Program Library Handbook
Hewlett-Packard, Software Center, 11000 Wolfe Road
Cupertino, CA 95014
(P,S,G,1972)

H.P. Educational Users Group Porgram Library Vol, I,II,III
Hewlett-Packard, Software Center, 11000 Wolfe Road,
Cupertino, CA 95014
(P,S,G,1972)

Huntington Computer Project
Polytechnic Institute of Brooklyn, 333 Jay Street
Brooklyn, NY 11201
(S)

Instructional Simulation Design
P.O. Box 3330
Leon Station
Tallahassee, Florida 32303

Public Policy Research Organization
University of California at Irvine
Irvine, CA 92664

REACT Curriculum Units
Northwest Regional Educational Laboratory, 710 S.W. Second Ave.
500 Lindsay Building, Portland, OR 97204

Secondary School Publication
Kiewlt Computational Center
Dartmouth College
Hanover, New Hampshire 03755
(P,S,G)

Simulation and Games
Sage Publications, Inc.
275 S. Beverly Drive
Beverly Hills, CA 90212

Simulation Councils, Inc.
P.O. Box 2228
La Jolla, CA 92037

Tecnica Books
Tecnica Education Corporation, 3301 Vincent Drive
Pleasant Hill, CA 94523
(P)

CONDUIT

Conduit is a cooperative experiment in transportability of curriculum materials. This National Science Foundation funded project uses Computers at Oregon State University, North Carolina Educational Computing Service, Dartmouth College, and the Universities of Iowa and Texas. Write to:

Computer Center
Oregon State University
Corvallis, OR 97331

Conference Reports

Conference on Computers in the Undergraduate Curricula. (conferences were held in 1970, 1971, 1972, 1973. These reports are an excellent source of ideas for the instructional use of computers at the college level.)

Newspapers and Newsletters

Computer Conversations
The Math Group, 5625 Girard Avenue South
Minneapolis, Minnesota 55419
(A collection of Math Problems for J.H. & H.S. problem solving)

Educational Products Group, Digital Equipment Corporation
146 Main Street
Maynard, Massachusetts 01754
(Ask for Educational Help Newsletter)

People's Computer Company
c/o Dymax, P.O. Box 310
Menlo Park, CA 94025
($4.00 per year for 5 copies of PCC Newspaper. Back issues are available. P,S, G)

Simulation/Gaming/News
Box 8899
Stanford, CA 94305

The Two Year College Mathematics Journal
53 State Street
Boston, Massachusetts 02109
(Published twice a year)

Periodicals

American Journal of Physics
American Institute of Physics, Inc.
335 East 45th Street
New York, NY 10017
(12 times a year, $18.00)

P - Prepared Programs
S - Simulations
G - Games

Journal of Educational Data Processing
Educational Data Systems Corp.
c/o D. Levin, Box 2995
Stanford, CA 94305
(4 times a year, $11.00)

The Mathematics Teacher
1201 Sixteenth St. NW
Washington, D.C. 20036
(Published 8 times a year)

SECTION C: PACKAGED PROGRAMS

by Jan Moursund and Wally Waldman

It is possible to write computer programs in a form so that they can be used by someone who does not know how to program a computer. Such programs can be stored on an auxiliary storage device (disk, magnetic tape, paper tape, punch cards) and brought into the computer memory when needed. It is common to refer to these programs as "packaged" or "canned" programs. A library of packaged programs is a critical part of the software of a computer installation.

Any computer program which is made accessible to someone other than the original program writer might be classified as a packaged program. In this article, however, we are concerned primarily with programs designed to solve specific types of problems. Thus we are not concerned with compilers, assemblers, or other major items of systems software. Nor are we concerned with CAI (compuper assisted instruction) systems. Typical "candidates" for inclusion in our category of packaged programs would be a program to solve a linear system of equations, a program to compute an orbit of a planet given several observations on its location, or a program to compute the correlation between two sets of measurements.

In order to consider the use of packaged programs in education, it is necessary to look first at a broader question: to what degree is it valuable to relieve the student of the necessity of actually manipulating his data--of "getting his hands dirty" with computations. One might argue for instance, that children no longer need learn multiplication tables, but instead should use electronic desk calculators to solve simple arithmetic problems; few educators, we suspect, would accept such a proposition. On the other hand, it makes sense to many teachers to allow students to use square root tables, desk calculators, or slide rules, rather than go through the labor of extracting square roots by hand. It seems clear that there are certain classes of problems which students need to learn to handle without mechanical help, and that there are others in which the availability of an "answer" is considerably more important than the means by which that answer was obtained. The line between the two is fuzzy and at times all but invisible. Nevertheless, the two categories do exist. "It is useful to distinguish between day and night", the statistician S. S. Stevens comments, "despite the penumbral passage through twilight." Recognizing that there is indeed a "twilight area" we shall in the next few pages attempt to restrict or attend to the class of problems for which access to solution is more important than mechanics of solution. In the context of this class of problems packaged programs represent some truly exciting possibilities for education.

The student who wishes to use a packaged program must deal with three sets of skills: selection of the proper package, preparation of input materials, and interpretation of output. Each of these sets has some important implications for the learning process, and we shall consider each in turn.

Selection of the Proper Package

In order to choose an appropriate program to solve a given problem one must first recognize the problem fits the problem class represented by the program. In other words, one must be able to generalize from a specific problem to a general model--one must be able to find the common dimensions or characteristics held by one's particular problem and the whole class of similar problems to which it belongs. Learning to recognize these relationships often requires a clear understanding of the nature of the problem and the logic of its solution; developing such an understanding is frequently much more beneficial to the student than learning to work his way through a "cookbook" solution.

To illustrate the concept of problem class we will consider quadratic equations. The general equation is of the form $ax^2+bx+c=0$ where a, b, c are specified in a given problem. A relatively small subclass of such problems consists of the cases where a=1 and the solutions are integers. Thus one could teach a unit on solving such equations as $x^2-5x+6=0$ and have students learn techniques (try each factor of "c") which are especially suited to this small problem class. We know, of course, that it is possible to give techniques to solve any quadratic equation. Completing the square, and use of the quadratic formula are two such techniques. But quadratic equations are themselves a subclass of all polynomial equations. Polynomial equations are a subclass of all algebraic equations. Algebraic equations are a subclass of all nonlinear equations in one unknown. It might be that one would want to teach and/or discuss methods for solving a general nonlinear equation, and to teach students to recognize when a problem falls into this category. This, in conjunction with a packaged program to solve nonlinear equation might be more valuable than memorizing the quadratic formula and developing skill in its use.

Consider for a moment the effects of incorporating packaged programs into a high school physics course. There is a program called DECAY2 in the Hewlett-Packard library that can handle a variety of problems dealing with nuclear decay. Students studying this phenomenon may use geiger readings, measures of time and mass, or the like to gain an understanding of what is happening. With access to the DECAY2 program they can look at many more sets of data than would otherwise be possible, and at the same time can focus on the nature of the relationships involved rather than on the details of arithmetic computations.

Preparation of Input Materials

This skill set breaks down into two main parts: learning to read and utilize program documentation, and learning to physically organize and prepare one's data.

Prepared programs of the sort found in computer center libraries usually are accompanied by documentation: written accounts of what the program intended to do, how it works (in varying amounts of detail, depending on the program) and instructions for inputting data. While documenters make every effort to write these accounts clearly and simply, it is inevitable that they develop some formalized terminology. Trying to work one's way through the documentation of a complicated program in itself can lead to a better understanding of and new insights into the solution process.

Organizing one's data carefully requires a certain amount of self-discipline; here too, the very process of organizing will often help the student to understand, at an intuitive level, the nature of the problem. Particularly in problems where large amounts of data are common, organization for computer analysis makes the student painfully aware of the implications of missing or mis-labeled information. Moreover the demands of the input devices of the particular computer system being used may provide the student opportunity to familiarize himself with the teletype terminal, the keypunch, the cathode ray tube, the paper tape punch, the optical scanner, and/or the other auxiliary machinery. In addition to being a valuable part of his education in and of itself, working with these sorts of tools is usually very interesting to the student and serves as a strong motivator for working with the problems at hand.

Interpretation of Output

A part of the documentation of each program will indicate the form in which solutions are output by the program. Sometimes this output is very simple and straightforward. The output device may simply print "THE ANSWER IS...", for instance. Other outputs may be complicated graphs, charts, and/or tables. Learning to interpret and make sense of the output regardless of its complexity, is again an adjunct to understanding the underlying nature of the problem.

In a broader sense, we may think of using the output of a program as a means of understanding the relationships among the variables dealt with in the problem. Programs can be run over many sets of data, in order to demonstrate the effects of manipulating one or more variable. In the nuclear decay program, for example, the number of radioactive particles in a sample may be studies over varying lengths of elapsed time and original mass. A financial program BANK (Hewlett-Packard library) can be used to show how a savings account balance is affected by interest rates, withdrawals, and deposits, and frequency of compounding. Another widely available program, PHOSYN (Hewlett-Packard library) demonstrates relationships between carbon dioxide concentrations, light intensity, and photosynthesis in plants. Again, it should be emphasized that the major gain to the student is not the solutions themselves, but the opportunity to study many sets of solutions and work out an understanding of the relationships reflected in them.

Sources of Programs

The bibliography following this section lists a number of sources of programs. Additionally most computer installations have program libraries; the potential user should check with nearby installations to find out what is available locally. One major advantage of using locally available programs is that one is usually assured that the program is compatible with the local computer system. The teacher need only run the program to make doubly sure it is suited to his students' needs (and that he or she knows how to use it) and he is ready to go.

Sources of Programs

A major failing of the computer science field is its continued re-invention of the wheel. For any simple problem solving application that one can readily think of, probably thousands of different packaged programs have been written. Almost for sure a program to solve the problem is available in a half dozen different computer manufacturer's program libraries, and hundreds of different computing center libraries. The point is, look before you program! Look first in your local computer center's library. Look also in the private libraries of your fellow teachers and/or other friends.

If you must look further, then the task becomes more difficult. If a program is in a library not on the local computer system there will usually be changes that must be made before it can be implemented on that system. Unfortunately, the variety of computer languages and hardware now in use frequently causes serious difficulties in shifting a program from the library at one institution to the library of another. For example, BASIC is not a standardized language. This means that different computer manufacturer's versions of BASIC differ considerably. A BASIC program that runs on the Hewlett-Packard 2000-C will probably need modification to run on a PDP-10 or PDP-8.

INFORMATION RETRIEVAL

by Cliff Burns

Introduction Society as we know it is based on information storage and retrieval. Even before the development of natural languages one person could show another how to do something (how to hunt) and that information could be passed down from generation to generation. The development of speech greatly facilitated the acquisition and transfer of knowledge. But the human memory is fallible, and much important information may be lost with the death of a key individual. Thus the development if pictorial methods of storing information, and eventually writing, were critical.

A tremendous step forward occurred as a result of the development of the printing press. Multiple copies of important information could be published and widely distributed. Literacy for the masses became a possibility. More and more people were not constrained by only having available the information that they could store in their heads.

The past 100 years have seen an ever quickening pace of human acquisition of information, and of development of methods and machines for storage and movement of information. Now radio, telephone, telegraph, and television allow the rapid movement of information over vast distances. Information is stored on paper, film (including microfilm) and in many computer reducible forms such as punched cards and magnetic tape.

It took about seven years to process the United States census information gathered in 1880. During that time it became evident that increases in population from 1880-1890, plus an increase in the amount of information it was desired to collect, would mean the 1890 census data would not be processed prior to time to begin the 1900 census. Thus there was strong motivation to develop new, faster methods for storing and processing such information. Herman Hollerith developed the punch card and associated unit record equipment for its use. Using his equipment, the 1890 census information was processed in just three years!

The U.S. government continued to be a very large user of automatic data processing equipment. When computers first became commercially available (1951) the Census Bureau was the first customer. Computerized information storage and retrieval has made rapid progress in the past 20 years; overall, however, the field is still in its infancy! This article discusses some of the key ideas in the field.

What is Information Retrieval?

Intelligent information in the form of handwritten or printed matter can be collectively described as records. In a library, some of the common records that we encounter are the books, periodicals, and the cards of the card catalogue. A record that most educators are familiar with is the locator card which is a condensed version of the data contained in the permanent record (such as name, parents' names, address, phone number, class schedule, etc.) which are kept on each student enrolled in school. Records can be gathered into organized collections from which information may be extracted as the need arises. The field of information storage and retrieval is concerned with the methods of creating and managing collections of records to facilitate the recovery of pertinent records. The use of information storage and retrieval systems is an everyday experience for almost everyone. Some examples of commonly used systems are: the library, correspondence files, checking accounts, telephone directories, and even the dictionary. As each of these systems grows in size, a point will be reached at which the manual process is not fast enough and automated systems are considered. The development of an automated information storage and retrieval system not only demands the development of technology and techniques for storing and manipulating records, but also the improvement of our understanding of the ways that people make associations and value judgments, and better methods of predicting what the information needs will be. Because of the cost of automating an information storage and retrieval system, one must not only meet today's needs but must predict with good accuracy what the needs of the user will be five years from now.

Operationally, all information storage and retrieval systems employ three basic processes: the analysis of records, creation of new records from old records, and the movement of records over distances. The central element in the system is the analysis, because the creation of new records and the transmission of records if determined by this phase. Analyzing records is the action of comparing a record or part of a record (sometimes called a key) with something else such as another record or a set of keys or features. A satisfactory comparison implies the ability to recognize important parts of a record. This is probably the most difficult exercise to communicate--from one person to another, let alone to a machine.

An example will help to clarify the key ideas of information storage and retrieval: A superintendent asks a principal for a report on the attrition (dropout) rate for his current graduating class. He wants to know what happened to large freshman class of four years back. Some of these are now graduating; others left the district, others dropped out of school, and many of this year's graduating class transferred to this school after their freshman year. Preparing this report will require sorting out all students who began as freshmen in the school four years ago.

in the school four years ago. Thus, the student record data base would be sorted using as a key the requirement of being a freshman in the school four years ago. The records of this group of students would then be sorted on the key -"moved out of the school district". The remaining set of students are the students who could possibly be graduating. Sorting this list on the key "is graduating" gives the data needed to determine the attrition rate.

An information retrieval system is created in anticipation of the need to retrieve certain information. Thus it is carefully planned and organized to facilitate the retrieval of certain information. The keys that one may wish to sort on should be decided in advance and the information that is stored in the system should be indexed using those keys. In a large sense this is like indexing a book. Certain topic headings are selected for the index; page numbers of each reference to these topics are given in the index. Information on a topic is retrieved by looking in the index, noting the page references, and then going to the apprpriate pages. These same ideas carry over to the computerization of an information retrieval system.

There are two standard strategies employed in developing an information storage and retrieval system. One method is to analyze and organize the information in anticipation of certain specific questions. As an example, student locator cards would be filed alphabetically by last name; this would be good for finding where a student is at a certain time, but would be a poor organization if one wanted to make a list of all students in Mrs. Jones' 4th period class.

A second organizational strategy is to avoid almost all preprocessing of the records. For example, the student locator cards could be in random order and when a question about the records is asked, the entire information file could be searched, one record at a time. This sounds inefficient, but can be done effectively by the computer.

In reality, most systems employ a blend of the two pure strategies. There is no reliable quantitative guideline for selecting the best mix of these strategies for a particular system. The role of the system planners and operators is to find an appropriate blend of processing and service activities that will give the best results.

A typical example of an information storage and retrieval system which blends these two strategies is a student record system. When a person wishes to find out where a particular student is during a given period, he wants quick access to the individual's record. Searching record by record would be a slow process, but if the records are in alphabetical order the search (using the name as the key) can be performed with little wasted effort. When is time to prepare a list of the freshman class, the search of all the records using the current class as a key, would select only the freshmen and would produce the names in alphabetical order. In a school of 2000 students (about 500 freshmen) which had an automated system, the above list could be produced in 10 minutes; but in a school with only 100 students (about 25 freshmen) the list could be produced manually in ten minutes and the increased speed of a computer would not be worth the cost of automating such a system.

Examples of Some Existing Information Systems

This section discusses four representative education oriented information retrieval systems.

1) Administrative--Oregon Total Information System (OTIS)
2) Research--Educational Resources Information Center (ERIC)
3) Instructional--Interdisciplinary Machine Processing for Research and Education in the Social Sciences (IMPRESS)
4) Library--Oak Ridge National Library.

OTIS--the Oregon Total Information System--was started in the mid-1960's by a group of Oregon school superintendents, under the direction of Dr. William P. Jones, the Lane County IED Superintendent. Financial support was obtained from the U.S. Office of Education. It is a cooperative computer center offering a range of data processing services to educational institutions. Actual operation began in May 1968. Steady growth has expanded the services to 250 schools with more than 100,000 students in 50 school districts, and several community colleges and educational agencies in Oregon.

The organization currently operates on receipts from member school districts. The basic charge for administrative services is $8.80 per student per year. Four times the volume of services, with lower expenses and smaller staff, are now provided, in comparison with the first year of operation.

Although not a state or county agency, OTIS is organized under the legal authority of the Lane County IED. Guidance is provided by an advisory board made up of educators from throughout the state.

The OTIS system is a large scale educational management system. It supplies student services such as those described earlier as well as enrollment, attendance, mark reporting, student class scheduling, and test scoring. In addition, it supplies business services, such as fiscal accounting, payroll, etc. This system can provide educational administrators the ability to retrieve from their districts' or schools' information file the data necessary for decision making and planning.

Some typical questions an administrator may ask of this system are:

a) What will be the cost of teacher retirement next year?
b) Precisely what will be the financial effect if teachers are given a three percent increment?

c) What was the grade frequency distribution in last year's standardized tests in mathematics? For the last three years?

ERIC is an example of what is generally termed a document retrieval system as opposed to a data retrieval system such as OTIS. It is like a massive library of all current material in the realm of education. It was created under the sponsorship of the Department of Health, Education, and Welfare. ERIC was created in answer to the information explosion in education amd the concern of educators for the burden of information dissemination. In the words of Lee Burchinal of the National Institute of Education:

> "We believe the day is not far distant when the ERIC network will link universities, professional organizations, school systems, boards of education--the entire educational community--to speed all research results to places where they are needed and when they are needed. That is our goal."

Through a collection of eighteen clearinghouses, each responsible for a particular portion of the entire educational area, documents are classified by a set of terms called descriptors which describe the document to those who wish to locate it. The classification of documents is one of the main functions of the clearinghouses and at each one there are experts on its particular educational area. The University of Oregon is one such clearinghouse and handles the area of educational administration. The descriptors are selected from a thesaurus which specifies for each entry one or more synonymous categories or concept classes. Each of these categories may be classified as a narrower term (NT), broader term (BT), or related term (RT). Currently there are about 15,000 entries in the ERIC thesaurus.

READING ABILITY 440
BT Ability
RT Cloze Procedure
Informal Reading Inventory
Reading
Reading Achievement
Reading Comprehension
Reading Development
Reading Diagnosis
Reading Level
Reading Skills
Reading Speed
READING ACHIEVEMENT 440
UF Reading Gain
BT Achievement
RT Academic Achievement
Early Reading
Reading
Reading Ability
Reading Development

Reading Level
Reading Skills
READING ASSIGNMENTS 440
BT Assignments
RT Reading

A search is begun by establishing a list of descriptors which will best collectively or separately describe the area of educational literature the user wishes to examine. The thesaurus is the key in establishing this list because only terms used in it will be used as descriptors. The precision of your retrieval is highly dependent upon a good selection of descriptors and also that the thesaurus can adequately describe each field of education. In addition to obtaining a list of references, the user may also request microfiche ($.65 per reference) or hard copies (Xerox at $3.29 per hundred pages) of the listed references. This search may be done manually or by computer, for a fee. To perform the search manually one must locate an educational institution which has all of the ERIC documents, (such as one of the eighteen clearinghouses). To have the search performed by the computer one must send his request to an authorized agent of ERIC; in Oregon this would the State Board of Education or one's local IED office. Either manually or by computer the user also has access to an abstract of the listed article so he can more clearly see whether or not the article is relevant.

In the instructional area there are few information storage and retrieval systems. IMPRESS is rather an elegant system developed at Dartmouth College during 1969-71. As surveys of relevant data are collected by social scientists they may be placed into IMPRESS files (a file is a collection of records, i.e., questionnaires). A codebook is prepared for each file, giving a name to each of the items that have been collected in the questionnaire, such as:

a) age
b) race
c) political party
d) which presidential candidate voted for in 1968 election
e) etc.

In addition, all possible responses for each of these items are provided so that the user knows the meanings of the coded responses. This is the storage part of the information storage and retrieval system.

To perform a search (retrieval) the user goes to the appropriate codebook and notes what variables (names) are used to refer to specific questions in a survey. After the data has been selected from the codebook, the user locates an available teletype. Next he identifies himself to the computer and selects the IMPRESS system. The IMPRESS system will then ask for data such as: which file (survey), which variables (items), and finally, what is to be done with the file. The IMPRESS system is more than just a retrieval system; it is also a small statistical system with a built in report generator.

IMPRESS is a data retrieval system which may be viewed as a rigid pre-allocated system similar to a student record system or a fairly loosely structured system like ERIC, wherein each record is examined during each search.

To better understand how a system such as IMPRESS may be used by teachers, let us view the following hypothetical classroom setting:

A class of modern problems at the East Podunk High School is studying voting patterns of people to determine what might be a good criterion for voter prediction. After a lengthy classroom discussion many different suggestions have been brought out, such as: age, income level, region, occupation, education, religion, etc. One student noted that a survey, sampling 1673 person nationally and asking 533 questions, was available to test these different hypotheses on the IMPRESS system. The teacher then gave an assignment, with many groans, to the students to check out different sets of hypotheses.

Later that day one could see the students poring over the PRES 68 codebook (PRES 68 was the name of the survey mentioned in class). They were finding voting preference is measured by an IMPESS variable called VOTED, religious preference by RELIG, occupation by OCCI, etc.

After the students had made their notes they went to the teletype room and signed on to the computer system in the usual manner and proceeded to generate the different cross-tabulations of variables to get at possible relevant facts. A command such as XTAB: OCCI, VOTED, produced the following output:

```
OCCI      BY     VOTED
DOWN:            RESPONDENT'S MAIN OCCUPATION-PBL  REF  CODE
ACROSS:          VOTED OR WOULD HAVE VOTED FOR WHOM IN 68

                        PERCENTAGES ACROSS

                 HHH         RMN          TOTAL

WHITE            45%         54%           423
BLUE             60%         39%           407
TOTAL            53%         47%           830
```

This command asks the machine to give the percentage preferring Humphrey and Nixon among white and blue collar workers.

The following day all the students returned with endless strips of teletype output to argue that their set of hypotheses was the best set of predictors.

This was, you realize, just a hypothetical situation but the actions taken are typical of users of the IMPRESS system.

An example of a fairly typical automated library system is one located at the Oak Ridge National Laboratory. Each borrower is issued an identification card similar to a credit card. In addition, a card for each book, bearing an identification number is kept with each book. When a book is borrowed, both cards are placed in a reading device which transmits them to the computer. Similarly, when the book is returned a message is transmitted to the computer indicating that the transaction has been completed. Late notice are prepared automatically. Reports to the librarian (management), on borrowing traffic and the precise holdings of each borrower, are prepared routinely.

Some of the newer ideas are to keep profiles of users and when new books arrive to notify the interested users. In a similar way, departments can be kept up to date as to the library's holdings which pertain to them.

One major impediment to the great use of computers in the library has been that as automation has been attempted, librarians have discovered that they know very little about users needs or how the users actually make use of their facilities.

What Does the Future Hold?

The future will see a greater use of automated information storage and retrieval systems. This will occur because of advanced techniques which are making the implementation of ISR systems easier. They are also going to be more economically feasible as the price of mass storage devices comes down and speed increases. In education these two changes, advanced techniques and lower mass storage costs, are the principal burden holding back educators' use of this most important concept.

SECTION E:
COMPUTERS AND JUNIOR HIGH
SCHOOL MATHEMATICS
by
Fred Board

Computers are already having some effect upon junior high school mathematics. For example, some of the content of "new math" was inspired by computers. A more down to earth example is the increased emphasis upon flow charts and flow-charting found in many modern texts. And finally, of course, some junior high school books actually contain material on computer programming, and many at least make mention of computers.

Before computers can have a really significant impact upon a mathematics curriculum, however, the teacher must have some knowledge of computers and "computer math." It is also helpful if students (or minimally, the teacher) have access to a computer. Under these conditions considerable change in the mathematics curriculum will occur. For example, Koethke (3) states: "Topics related to algorithm and computation can no longer be treated as supplementary; they have become a very necessary and important part of the mathematics curriculum."

A major goal in both mathematics and in computer science is problem solving. To use a computer to solve a problem it is necessary to have an appropriate computer program. Often such programs can be written by the students in a class, and this writing process can be a valuable part of the course. Before a program can be written and used successfully three main objectives must be met: 1) the problem must be understood. 2) An algorithm or procedure must be developed to solve the problem. 3) Test data must be prepared to determine if the process actually does what it is supposed to do. Once a student has met these three objectives, he is well on his way to having a successful program. Any changes in the mathematics curriculum should reflect these three objectives as well as Koethke's comment on algorithms.

<u>MATHEMATICS GOALS AND THE COMPUTER</u> Any change in the mathematics curriculum must either change the goals and objectives of mathematics education or be implemented to help reach the present goals. Listed below are six major goals of math education as stated by Johnson and Rising (2). Following each objective are some suggestions as to how a computer may help the students reach these goals. After the computer becomes a more familiar instrument in the classroom, some goals may be changed or new ones added.

1. "The student knows and understands concepts such as mathematical facts, or principles." Included in this objective is the ability to apply knowledge to a new situation. The computer opens up an entire area of exploration and certainly offers an opportunity to apply newly learned skills.

2. "The student understands the logical structure of mathematics and the nature of proof." In Algebra I, a student learns to find the zeros of a function by a variety of methods, such as factoring or using the quadratic formula. By implementing the computer, Newton's method of solving equations can be taught and the student is able to find the zeros of almost any function. Using methods like these should improve the students understanding of the structure of mathematics. Students can learn to solve classes of problems rather than employ special "tricks" to solve certain types of problems within the class.

The concept of proof, or the correctness of a chain of arguments, is quite difficult for many students to learn. Indeed, many fail to appreciate the need for proof. A computer program is, in a large sense, a proof. Many of the concepts of proof can be taught in the process of teaching computer programming. A student who can write a correct program to solve a class of problems has developed many of the skills needed to prove a theorem, and in a context that may be more acceptable to the student.

3. "The student performs conputations with understanding, accuracy, and efficiency." Much of mathematics education seems to emphasize accuracy in computation, even to the detriment of understanding. Machines can do computations more rapidly and accurately than people. What is left is the need for understanding--the use of machines places considerable more emphasis upon understanding. The emphasis on student computation will be one of the areas that will have to be re-evaluated as machines are used more. If machines are available, should we expect students to be able to do long involved arithmetic computations, or will approximation be sufficient? Approximation will be discussed below.

4. "The student has the ability to solve problems." With less time being spent on actual computation, and more on problem solving, students should be able to move on to more difficult problems more rapidly.

5. "The student develops attitudes and appreciations which lead to curiosity, initiative, confidence and interests."

6. "The student learns how to develop proper methods of learning mathematics and communicating mathematics, and also develops study habits essential for independent progress." These are grouped together because they are best answered by Stenberg and Keotke. "Aside from computing, I have seen no field of study in which so many students become inspired to invent problems, learn on their own and tackle major projects." Stenberg (4). When refering

to the alteration in the role of the teacher when computing is used in conjunction with math, Koetke (3) stated: "Most apparent is that the time spend reminding students that they must pursue other studies in addition to mathematics far exceeds the time spent in pursuit of students who are not attempting to complete their work."

AREAS OF SUGGESTED CHANGE In the text below are some suggestions for change in the 7th and 8th grade math programs. These are written under the assumption that programming is introduced at the 7th grade or earlier, although most of the changes could be implemented without access to a computer. This is not intended to be an enhaustive list; rather it is a partial listing of suggestions.

First let me give three programs in BASIC that might have been written by a 7th grader. They each illustrate ideas to be discussed later.

```
10  REM TABLE OF SQUARES AND SQUARE ROOTS
20  LET K=1
30  PRINT K, K↑2, SQR(K)
40  LET K=K+1
50  IF K<=100 THEN 30
60  END

10  REM AREA AND PERIMETER OF A RECTANGLE
20  READ L,W
30  PRINT L, W, L*W, 2*L+2*W
40  GO TO 20
50  DATA 16,7
60  DATA 23.8, 14.9
70  DATA 62.97, 62.97
80  END

10  REM COMPUTER DECIMAL EQUIVALENTS OF FRACTIONS
20  READ N,D
30  PRINT "THE COMPUTER DECIMAL EQUIVALENT OF"; N;"/";D;"IS";N/D
40  GO TO 20
50  DATA 2,3
60  DATA 1,7
70  DATA 2,7
80  END
```

Study of Variables

The concept of "variable" is one of the most important ideas in mathematics. It is also fundamental to computer programming, as is demonstrated in even the most simple programs. "Traditionally, a variable is a symbol for which values may be substituted from a certain set called the domain of the variable. In computing a variable is a symbol which at any time has a definite value although this value may change from time to time." Stenberg (4).

Teaching variables as having a definite value at any time seems to help students understand expressions such as (-B +√B·B - 4 · A · C)/(2 · A). Each variable has a value and the expression becomes a means of calculating a numerical value rather than a string of symbols to be memorized.

Algebraic Symbols Man creates symbols and assigns them meanings. The same symbol may be assigned different meanings in different contexts. Thus a "period" is used to denote the end of a sentence, as a decimal point, and as the symbol for multiplication. The "equals sign" in programming has a distinctly different meaning than the same symbol in algebra. In programming X=X+5 is a perfectly reasonable "assignment" statement, indicating that X is to be increased by 5. In algebra this is an equation with no solution, or an example of a nonsense statement. To remove the possibility of misunderstanding, some programming languages use a different symbol, such as := in their assignment statement. Similarly, ← is often used in the assignment statement in a flow chart.

A student should be exposed to the above ideas, and should learn to cope with a variety of notational systems. He should get some insight into good and bad notation, and the value of having a good notational system.

Flow Charts

An algorithm is "a complete, unambiguous procedure for solving a specified problem in a finite number of steps." Dorf (1) p. 41. A computer program must also contain the characteristics mentioned above: a. Be finite (have an end to the number of steps) b. Be complete (do the job) and c. Be definite (the computer must be able to do each step).

If an algorithmic approach to math is implemented as suggested above, then a natural way is to use flow charts. A study reported by Stenberg (4) showed that the use of flow charts in math classes greatly increased the students performance when compared with a control group not using flow charts. This study was conducted on junior high students in a Minnesota school district. A flow chart is a diagramatic representation of an algorithm. This step by step process is easily learned by students and quickly adapted to problem solving.

The use of flow charts is an easy way to begin teaching an algorithmic approach to math. In addition, flow charts are easily converted into programs. Once a student is able to flowchart a program, most of his work in writing a program for the same problem is finished.

Fractions and Decimals

When was the last time you saw a problem like 2/7 + 1/9 or 3/14 X 21/6 outside of a math textbook? If you answer honestly, it was probably a long time ago, if ever. Most people in the "real world" don't have many problems like this and the ones who work this type of problem generally have a calculator or computer at their disposal.

A computer does a complicated computation such as 87.93 times 164.37 just as rapidly and easily as it computes 2.5 times 3. Similarly it has no trouble handling computations such as 2/7 + 1/9. The need for a student to develop great skills in doing such computations by hand is questionable. Does this mean fractions and decimals shouldn't be taught? No, but it does suggest that a change in emphasis is appropriate.

One possible change would be to place greater emphasis upon approximations, and developing an "intuitive feel" for the relative sizes and inherent "meaning" of various fractions and decimals. Some ideas related to this are discussed in the section on Approximation which follows.

Another important idea is that of scientific notation. In the language BASIC all computations are done in floating point arithmetic, which is a machine version of scientific notation arithmetic. The floating point number line is quite interesting, and distinctly different from other number lines (integers, reals) that the students will have previously studied. It is a source for many interesting and difficult questions.

Approximations

If a student wants accurate answers to a complicated computational problem then he should make use of a tool such as an electronic desk calculator or a computer. Most "real life" problems don't require accuracy, however. That is, the housewife, or logger need good skills in computing "approximate" answers to relatively simple problems. The ability to make good estimates and appropriate approximations, and to carry out simple computations in one's head, is valuable to most people. More emphasis could be placed upon this in mathematics education if less emphasis were placed upon developing skill in doing more complicated computations.

The ability to make appropriate approximations and to do rapid (simple) mental arithmetic is also quite useful in the computer programming field. One of the major aspects of writing a program is to make sure that it is logically correct. A standard approach is to prepare test data, and to compare the computer's answers with one's own hand computed (or, computed via desk calculator, or in one's head) answers.

Gross errors in a computer's output are often obvious to the person who has good number sense--i.e., to one who has a good "feel" for what an appropriate answer is.

SUMMARY So far computers have had relatively little impact upon junior high school mathematics. It should be evident to the reader that there are a number of possible changes that might occur because of computers. Currently, however, few teachers have the knowledge to implement such changes, and few students at this grade level have access to computers.

Some of this situation can change (will change?) quite rapidly. Electronic desk calculators are now quite cheap, and could become a standard tool in the classroom. The larger change, having computer terminals in every math classroom, will be longer in coming. But, it is coming!

Bibliography

1. Dorf, Richard C. Computers and Computer Science, San Francisco, California: Boyd and Fraser, 1972.

2. Johnson, Donovan A. and Rising, Gerald R. Guidelines for Teaching Mathematics, Belmont, California: Wadsworth, 1967.

3. Koetke, Walter "The Impact of Computing on the Teaching of Mathematics". AFIPS Journal, Vol. 40, Spring 1972: 1043-1049.

4. Sternberg, Warren "Computing in the High School-Past, Present and Future-and its Unreasonable Effectiveness in the Teaching of Mathematics". AFIPS Journal, Vol 40, Spring 1972: 1051-1058.

SECTION F. STUDENT WRITTEN PROGRAMS FOR PROBLEM SOLVING

by Dick Bach and David Moursund

There are many possible approaches to the instructional use of computers. Currently one of the most common at the secondary school level is to incorporate some instruction in computer programming into a standard math course, and then have the students use their programming skills as they study the standard math topics. Such an approach can be used in any of the mathematics courses standardly offered by secondary schools. This article discusses a few of the key issues; particular attention is given to the "Colorado Project" materials, which are an excellent example of instructional materials incorporating computer programming into a standard mathematics course.

An initial question is "If a computer is available, why not just use packaged programs?" This is not an easy question to answer. In general one uses package programs when "getting the answer' is the goal. One does not do this when it is desirable for students to gain a clear and complete understanding of the process needed to get the answer.

It is generally accepted that if a student can write a program to solve a certain type of problem, then he probably has a good understanding of how to solve that type of problem. This is a key idea. A computer program is a detailed step by step set of directions telling a computer how to solve a certain type of problem. In general, telling a computer how to solve a problem is more difficult than telling a person, or actually solving the problem. A computer program is much like a proof. It consists of a logical sequence of steps, each of which must be correct, and which must be performed in a correct order if the problem is to be solved. To write such a "proof' is often very challenging to the student.

There are four things needed if one is to have students write programs as part of their activity in studying mathematics. These include a certain amount of programming knowledge on the part of the students, a larger program knowledge on the part of the teacher, computer access, and appropriate instructional materials.

The programming knowledge needed by students can be integrated into any mathematics course, with perhaps a week or so of concentrated effort on computing at the beginning. An important idea here is that of imitation. Suppose a student studies a program to compute the area and perimeter of a rectangle. (A BASIC example is given below.)

```
10   LET L=15.8
20   LET W=6.5
30   LET A=L*W
40   LET P=2*L+2*W
50   PRINT L,W,A,P
60   END
```

Using this as a model, almost all students can write programs to carry out the computations indicated by most of the standard formulas appearing in secondary school math course below the trigonometry level. When "undoable" cases occur, such as a formula involving square roots, additional language features (like SQR in BASIC) are easily presented to the student. The major point here is that one presents only a limited amount of instruction in computer programming. Each additional topic should be motivated by need for that topic.

As far as teacher knowledge isconcerned, it is perhaps enough to say that the teacher should know significantly more than he expects most students to learn. The teacher should be comfortable with using computers; he should be able to read and understand simple student-written programs as easily as he can read and correct other homework. (Note: the brighter, more venturesome teacher may disregard this suggestion. Learning by doing is common in the teaching game.)

Having appropriate computer facilities is important. However, quite a wide range of facilities can be used provided the course is adjusted to the facility. At the secondary school level (assuming computer facilities are available) one most typically finds one or two time-shared terminals connected to a remotely located computer. These may be located in one of the math classrooms, or may be in some other part of the building. If a teacher has access to only one time-shared computer terminal, and it is located in the administrative offices, he will probably have relatively little success in having his students write and run programs as part of their math activity!

A much more desirable setup is to have several computer terminals in the math classroom (hopefully, in a soundproof adjoining "office", with access from the hallway) or to have a single terminal and a mark sense card reader. In the latter case students write programs on mark sense cards and these are batched to the computer. Quite a few programs can be processed during one class period.

Finally, there is a need for appropriate instructional materials. By and large, relatively few such materials have been developed for use at the secondary school level. Of course, many hundreds of teachers have developed their own materials, and this will continue in the future. But on a nation-wide basis there is little agreement on what constitutes appropriate materials, and few texts have been published specifically aimed at the secondary school level. Two exceptions are the CAMP materials (2) and the Colorado Project materials (1). The latter is discussed below.

The Colorado Schools Computing Science Curriculum Development Project has prepared a text which incorporates BASIC language programming into second year algebra and trigonometry. A quote from the Preface describes the general nature of the mathematical content:

> "Most of the topics of the usual algebra-trigonometry course will be found in this text. However, the emphasis placed on various topics may be different from that normally given in other texts. The authors feel this is a result of an honest effort to uncover the proper role of the computer in mathematics education. In Chapters 1-3, properties of sets, axioms of the real number system, rules of proof, and programming are presented. These concepts are used in a precise development of the mathematics in the reamining chapters. The concept of a function becomes the unifying thread of Chapters 4-10." (1)

Topics covered in Chpaters 4-10 include relations and functions, linear functions, circular functions, quadratic functions, polynomial functions, sequences, and exponential and logarithmic functions.

There is considerable interest in the Colorado Project materials in Oregon. Marshall Watkins (Beaverton) and Bill Best (Willamette High School) ran a three day workshop on these materials during summer 1972 for teachers in the Portland area. A similar four day workshop, by Dick Bach (North Eugene High School) and Bill Best, was held in the Eugene area in summer 1973.

At the risk of closing on a negative note, it must be pointed out that problems can arise with this approach to teaching mathematics. If one does not have appropriate teacher knowledge, computer access, and curriculum materials, then the computer parts of the course may prove unsatisfactory. One must also face a standard argument--if time is spent teaching computer programming, then something else must be dropped from the course. If one's school, school district, and/or administration think of a course only in terms of the specific topics it covers, then this will be a considerable problem. A better way to approach this situation is to think of the overall effect of the course on the student. If the student learns as much or more, overall (remember, computer programming is a worth while topic in its own right) and learns it in an interesting and modern environment (using computers) , then it is difficult to argue against this approach.

1. Beavers, Collins, Herber and Larsen; Second Course in Algebra and Trigonometry With Computer Programming; The University of Colorado; 1973.

2. Johnson, David (Editor and author of several parts): Computer Assisted Mathematics Program (6 Volumes), Scott, Foresman and Company, 1969.

Section G: Abstracts of related articles for Chapter IV

Classroom Computers Serve Students
Nation's Schools, Vol. 83, March 1969
Pages 88-89

In Yorktown Heights, New York, a group of sixth grade students are participating in a project to determine if computer simulations are an effective teaching tool. The goals of instruction had to do with basic economic principles, such as how an agricultural economy works, how capitalism works, and economic problems of an emerging nation.

The student interacts with a sophisticated computer program through use of a standard teletypewriter. The computer also controls an audio-visual device which contains a slide projector and audio tape recorder.

The program simulates a total environment in which the student has a specific role, such as king of a small nation. Through a combination of role playing and programmed instruction, the student makes economic decisions relevant to his role, and sees the effects of these decisions.

A 6-weeks comparison of 26 students using the simulations with a matched control group in a traditional setting revealed the following: 1) all but two of the students with reading difficulties finished the units; 2) almost all of the students said they enjoyed the simulations; 3) in one unit the experimental group (computer simulations) showed significant gains over the control group, and on another unit no significant differences were found; and, 4) the experimental group used less student time.

Dave Dempster

Alpert, D. & Bitzer, D.L.
Advances in Computer-Based Education
Science, Vol. 167 No. 3925 (March 1970)
Pages 1582-1590

The quantity and quality of education possible when a school utilizes a high-speed computer is as yet an unanswered question. The PLATO system was installed at the University of Illinois for the purpose of 1) exploring educational possibilities of the computer in the instructional process and 2) designing an economical system that incorporates valuable approaches to learning. In connection with the first objective, four misconceptions are common: 1) computer-based education is synonymous with programmed instruction; 2) the instructional strategy must anticipate all conceivable student responses so as to compare them with "correct" answers stored in the machine; 3) computer-based instruction may be useful for the transfer of information but is not of value in the development of critical thinking; 4) a computer system used for computer-based education cannot be used in a time-sharing mode for conventional computer programming. The article refutes each of these misconceptions.

Just as the textbook has distinctly different uses at various educational levels, so must computer-based education. This article describes some alternate uses of the computer in education using PLATO. A comparison of students in a traditional medical class vs those in a computer simulated class utilizing computer simulation is given. Students using PLATO scored as well in medical diagnosis at the end of a semester but put in one-third to one-half as much time in instruction; over a half year period, the computer group retained more than the normal class group.

Available CAI systems entail costs which range between $2 and $5 per student contact hour. The intended PLATO IV system could reduce this cost to about thirty-five cents per student-contact hour.

The availability of a system that is economically feasible and educationally structured could produce some of the following: 1) reduction in the number of large lecture classes, 2) special instruction at home for physically handicapped students, 3) gradual abolishment of lock-step schedules and narrowly specified curricula in formal education, 4) effective job training and retraining, 5) provision of remedial instruction or tutorial assistance during regularly scheduled courses for students with insufficient preparation, and 6) provision for continuing education for professional personnel.

Overall this is an excellent article about one of the most exciting aspects of computers in education.

Lloyd Fraser

Ritter, Gary G.
Calculus and the Computer: An Evaluation by Participants
The Two-Year College Mathematics Journal, v1, n2, (Fall 1970), pp. 41-49.

This article discusses a questionnaire evaluation by the participants of a calculus-with-computers project which was done under the auspices of the Computing and Mathematics Curriculum Project (CMCP) during the fall of 1969. Five Colorado colleges and universities participated in the computer-extended calculus project.

Interested instructors were selected at each of the schools to teach a computer-extended introductory college calculus class. Each instructor was permitted to assign any calculus text he desired. The computer was used to demonstrate the various important introductory calculus concepts and to solve homework exercises. No previous programming experience was required. The programming language used was BASIC. Computer time sharing was used to accommodate the computer needs of the classes. Each school had one Teletype (input/output device) per twenty students. The capability of a time-sharing service to allow a student to input various guesses, get output immediately and to immediately input new guesses on the basis of this output was considered to be necessary for the incorporation of the computer into the calculus course.

Computer units were written by this author for distribution to the instructors of the participating colleges and universities. The units were written on the topics of function, limit, derivative, application of derivative, integration and numerical integration. Each unit included an explanation of the concept into the class, completed examples with computer programs and output, discussion of the example, and selected exercises that the instructor could assign the student. The units were presented to the class as the relevant topics were taken up in class.

An evaluation questionnaire was given to the students and the results and a discussion of them can be found in the article. Two conclusions of particular interest are: 1) students are aided by computer applications in organizing their thoughts for attacking problems; 2) the computer-extended topics of function, limit, and derivative were helpful for the students' understanding of these concepts.

Sue Waldman

Boblick, John M.
The Use of Computer-Based Simulations and
Problem Drills to Teach the Gas Laws
Science Education 56(1): 1972, pp 17-22

A fundamental part of a high school chemistry course is the study of the behavior of gases. Reliable laboratory equipment that would enable a student to experimentally determine the relationships among the volume, temperature and pressure of a gas usually is not available. A computer-based simulation has been devised so that a student can investigate the behavior of an ideal gas.

Initially, the student is told that at the end of the program he or she should be able to express the relationship between the volume and the pressure of a gas. After being informed of the goal the student interacts with the computer system to gather data. The investigation begins by entering the variables which remain constant during the simulation. In the case of Boyle's Law, the number of moles of gas and the temperature remain constant. The computer calculates the values of the corresponding dependent variable and displays for the student both the set of pressures which he or she entered and the corresponding volumes. The student than has the option to view a graph of the variables plotted against each other.

After viewing the data which she has just collected, the student may choose to go directly to the multiple choice quiz, continue to investigate the relationship between the variables, or see the teacher for help before continuing. Drill problems are built into the program to direct the student to investigate different properties of the gas laws. The student is able to control the drill and provide for himself the amount necessary to acquire the desired problem-solving skills. When the student completes the drill she is provided a summary of her performance; this summary also may be provided to the classroom teacher.

Students were given pre-tests and post-tests to measure performance. Their attainment of the terminal objectives of the programs exceeded expectations. On the portion of the test pertaining to the basic gas law relationships 90% of the students answered correctly 92.6% of the questions.

The programs are being reviewed to improve the response time by programming revisions and newer programming languages. An improved response time would enable a student to move through the drill much faster and solve more problems in a given time. Most student comments about the program were positive in nature but some remarked that the drill problems were presented too slowly.

Ron Boys

Dorf, Richard C.
Data Banks, Information Retrieval and Libraries
Introduction to Computers and Computer Science, Chapter 12, pages 351-370.
Boyd & Fraser Publishing Company, 1972

The book "Introduction to Computers and Computer Science" is an excellent example of the recently published texts designed for an introductory sequence in computer science at the freshman-college level. Such texts are also suitable for use with good high school students. Quoted below is a summary of one chapter.

"The computer is a useful device for information storage, processing and retrieval. With the advent of relatively low cost storage the computer lends itself to the development of data banks. A data bank is an on-line storage unit retaining large masses of data. Data banks are used to retain economic, social and industrial data among others. A terminal is one possible access connection for the potential user. A computer data bank is automated, of high speed, and it usually operates at a reasonable cost per item stored.

Information is the meaning assigned to the data or an aggregate representation of the data. It is the desired output of an information retrieval system. Information retrieval is the process of accumulating, classifying, storing and searching large amounts of data and extracting the required information from it. The quality of an information retrieval system is measured by how much relevant information, in comparison to how much irrelevant information, is provided in response to a query. Information retrieval systems have developed for airlines, government functions, and the legal profession among others.

Computer information retrieval systems will aid in the operation of the libraries of the future. Computers are being used for automated acquisitions, receiving, cataloging and circulation. The primary items of cost for automated libraries will be the amount of information stored and the response time required as a result of an inquiry. In the future, a device in the home of the office may display information upon request.

Computer information systems containing information about individuals are increasingly necessary to govern and manage our complex society. However, the ready availability of information about persons leads to a concern about the privacy of information. Privacy, in this context, is primarily concerned with unauthorized access to personal or confidential information. The conflict between the need for accumulating information in a data bank, and the protection of an individual's freedom, must be reconciled.

Sister Clare MacIssac

Dorf, Richard C.
Simulation and Games
Introduction to Computers and Computer Science pp. 371-425
Boyd & Freser Publishing Company, (1972), Ch. 13

Modeling and simulation are of great value to business, industry and goverament because they permit one to study the effects of various decisions or choices without going through the complete process of the phenomenon being considered.

A computer model is a representation of a system or phenomenon in a mathematical or symbolic form suitable for demonstrating the behavior of the system or phenomenon. Simulation involves subjecting models to various stimuli or situations in such a way as to explore the nature of the results which might be obtained by the real system.

A computer simulation may be developed in FORTRAN, BASIC or a language specifically developed for simulation. Two widely-used simulation languages are GPSS (General Purpose Simulation System) and SIMSCRIPT, which is a FORTRAN-based language

The digital computer has recently been introduced as a general purpose simulator in the high school classroom. The idea is to expose students to experimentation which would not normally be available in the school laboratory. Simulation is particularly apropos for students when the experiment is too complex, expensive or dangerous to carry out in the laboratory. A computer simulation program entitled EVOLG permits the user to explore some of the factors which affect evolutionary changes. In addition, a program entitled POLUT, which is an elementary simulation of a water-pollution situation, has been developed. Such uses of simulations in the schools clearly affords opportunities for students to experience certain phenomena rather than to learn about them vicariously from teachers.

A major advantage of the utilization of simulation techniques is that the participant can learn from his mistakes without suffering the real-life consequences when mistakes are made while learning. The simulation exercises are useful in teaching political and diplomatic skills when the consequences of error in the real-life context are so costly as to effectively prohibit trial-and-error learning.

Today computer scientists are dedicating a considerable amount of effort toward programming digital computers to play games. A game, which is a common activity, is a closed system with a set of explicitly stated rules and a fixed goal. A game has a rational, analytic component and an emotional, creative component. The game's analytic dimension includes the strategic and structural characteristics experienced in life. The emotional aspect of the game includes chance and a realization that often the action of the game is as important as the outcome. Games are quite useful in training students, managers and public officials to make wise decisions and to experience the resulting consequences. Also, computer games are of interest to computer scientists since the programming and development of algorithms of games is a challenge in itself.

Wally Waldman

Dorn, William S.
Computer-Extended Instruction: An Example
The Mathematics Teacher v63 n2 (February 1970) p147-158

This article discusses the computer and its use as a laboratory in conjunction with a traditional mathematics course. This type of computer use has a direct analogy with the use of a physics laboratory as it applies to a physics recitation class. The mathematics used in this article should be within the reach of most 8th or 9th grade students. That is, it uses only elementary algebra including some inequalities. However, the problems are interesting and difficult enough to challenge 12th grade students and, in fact, any pre-calculus student.

Some simple maximization and minimization problems are discussed and a computer is used to perform some experiments with them. The objectives of these experiments are: 1) to develop some insight into the mathematical concepts of maximum and minimum; 2) to allow the student to "guess" at some theorems regarding when a maximum or minimum occurs; 3) to motivate the student to study the calculus.

This article also discusses the value of performing some preliminary mathematical analysis in order to reduce the amount of computer time consumed and to reduce the amount of data that is printed. Perhaps even more important is the discussion of how and why we should have the computer do some bookkeeping work for us.

Some particular experiments are presented. The results of the experiments were used to make some conjectures. In one case the conjecture was actually proved. A lengthy discussion followed on just what constitutes a mathematical proof and what part a computer can and cannot play in a proof.

The article contains examples including maximum area and volume. Instruction is extended as the results of one program initiate another question until the reader is led one by one through several programs until the initial conjecture is found to be true.

Wally Waldman

Dowling, John Jr.
A "Canned" Computer Lab
American Journal of Physics, v40, (January 1973) pp 76-80

The use of a set of "canned" computer programs (Fortran IV) in an introductory mechanics course for science majors is discussed. This type of lab should be advantageous to the physics teacher who does not have access to an on-line or time-sharing system but who does have access to a computer which has a Fortran IV compiler.

The main purposes in using the computer were (1) to free the student from the grubby work of doing all the calculations and writing the lab report and (2) to enable the student to gather much more data in order to "see" the physics in the lab. The first objective was successfully achieved, but the second objective was only partially successful. There were three types of labs given during the semester: (1) all computer, (2) half-computer--half-student, and (3) all student.

A brief discussion of the utilization of a "canned" program for determining the coefficient of restitution of a glider on an air track is given.

Any lab that has long tedious calculations is best performed via computer because it enables the student to concentrate more on the experiment rather than the mathematics. Any lab that shows a trend developing, either evident or not so evident, should be performed by this approach; otherwise the student can not realistically be expected to perform enough experiments to see the trend. Much more effort can be put into the development of the theory of errors and the uncertainty associated with an experiment. Questions dealing with the lab can be interspersed in their appropriate place in the report. One bonus to the computer-printed report is that everything is in place (relatively, that is). This makes it easy to examine and correct.

Students today should know something about the computer: its possibilities, its limitations, and its applications. This is a good way for a cross section of a student population to be introduced to the computer.

Some concern was expressed for the student's frustration with Fortran IV's decimal and integer type number representations. If a "free style" input had been available this minor inconvenience would not have appeared. Teachers should keep the data input specifications simple and adaptable, and adequately define the computer's role in the "canned" lab.

In the final evaluation the "canned" lab approach did help some students and freed practically all from the drudge tasks associated with a lab and therefore was considered a moderate success and am improvement over previous methods.

Wally Waldman

Dwyer, T.A.
Teacher/Student Authored CAI Using The NEW BASIC System
Communications of the ACM, Vol. 15, (January 1972), pp. 21-28

The CAI system discussed in this article is undergoing development within a large urban school system (Pittsburgh) as part of an experiment in the regional use of computers for secondary schools.

Two modes of usage are defined. One is dual mode and the other solo mode. In the dual mode the student interacts with a pedogogically-intended master program. An example of this type CAI program would be a simulation. In the solo mode, the student writes, debugs, and revises an original program. It is suggested that student use of a system in the dual mode motivates them to operate in the solo mode.

The requirements for such a CAI system are: (1) easy system access in all modes, (2) availability of the full power of general purpose computing to all users, and (3) understandability of the system in terms of non computer-oriented teachers.

The working system uses a CAI processor called CATALYST (Univ. of Pittsburgh). The algorithmic language is an extension of BASIC called NEW BASIC (NBS). NBS extends BASIC in four areas:

1. Extension as an algorithmic language (i.e. suffix control: FOR, WHILE, ELSE etc.).
2. String functions allow for greater freedom in responding to questions.
3. Interactive capability similar to languages such as JOSS.
4. Users can write functions to suit their own needs.

The feature of NEW BASIC that is most attractive to educators is the mixing of dual and solo modes. A student, operating in the dual mode, may elect to temporarily go to the solo mode. He then returns to the dual mode at the point he left it. This allows a student to use his own program to calculate an answer for the dual mode.

In one test school, terminal usage increased over 100 connect-hours per month per terminal after one semester with the new system.

Robert Layton

Hammond, K. R.
Computer Graphics as an Aid to Student Learning
Science 172:903-8 May 28, '71

In computer graphics the student uses a console on which he sees information displayed pictorially (like a graph). With this handy aid, he can easily compare what should be done with what he is really doing. In this sense he is actually trying to apply his knowledge.

To apply his knowledge correctly to a given task, the student must be able to make many comparisons. He must be able to compare properties of his judgmental system with properties of the task he faces so an appropriate match can be found. The computer can help him to make such comparisons vividly, quickly, and accurately.

With computer graphics to aid in such comparisons, test trials with medical students at the University of Colorado did show that good judgment can be learned rapidly. Computer graphics also has possibilities for other students in either specialized or refresher courses. Research ingenuity may indeed find many methods of using computers to study human thought processes.

Don Weaver

Hickey, Albert E.
The Use of the Computer in Mathematics Instruction
The Two-Year College Mathematics Journal, v1, n1, (Spring 1970)
pp. 44-54.

The author lists and discusses five modes of student-computer interaction. They are: problem solving, programmed desk calculator, simulation, drill and practice, and tutorial.

Problem solving is by far the most common instructional use of the computer at present. The student is required to 1) analyze and structure his problem, 2) formulate a step-by-step solution to be carried out by the computer, and then 3) encode the solution procedure in such a way that it can be communicated to the computer. The article presents a set of problems that can be used to correlate problem solving with the precalculus mathematics curriculum.

The remaining uses of the computer in instruction require that the computer be preprogrammed to some degree. Whether the student writes his own program or draws it ready-made from the system library, in the end he has at his service a tool which he can use to expedite his investigation of the properties of mathematical functions and quantitative phenomena. This desk calculator mode of interaction may be even more valuable to the science student, to expedite processing of laboratory data, than it is to the mathematics student.

In the third mode discussed, the computer can be loaded with a program that as a) available in the library of the time-sharing service, b) written by the student, or c) purchased from a commercial source as a software accessory or "package." The program causes the computer to generate data characteristic of a physical process or mathematical function which might otherwise be inaccessible to the student. This capability is particularly valuable in the study of probability and statistics. The computer simulates a random process by resorting to a subroutine which is a random-number generator.

The fourth use of the computer presented is the drill and practice program that presents the student with a problem to which he must supply an answer. Drill and practice is distinguished from the desk calculator and simulation modes by the requirement that the student respond.

The fifth and last mode discussed in the article is the tutorial mode. In this mode of interaction, the computer simulates much of the behavior of the skilled teacher and may add some qualities even the skilled teacher may not possess, e.g. virtually unlimited memory for past performance of students, a store of instructional alternatives, an ability through high-speed calculation to let the student introduce data and see their effect on the output, etc.

Of the five instructional processes discussed in this article, the tutorial mode is found to be the most costly not only in terms of operating cost per student, but also in terms of the cost of preprogramming the lessons.

Wally Waldman

Kieren, Thomas E.
Computer Programming For the Mathematics Laboratory
The Mathematics Teacher V. 66, n 1, (Jan 1973)

The author suggests that a worthwhile lab would involve the use of a computer facility to "solve problems and even do some elementary 'research'". He illustrated with a seventh grade class that was interested in writing programs that would direct the computer to print the primes between 1 and 100.

> "In the process of writing programs, executing and debugging them, and studying program output, the students learned a great deal of mathematics about prime numbers."

Kieren goes on to discuss the benefit derived from the computer in individualizing instruction and gives a scheme for ordering exercises according to difficulty and educational function. Possible sources for problems are suggested as well as a format for student records.

In my estimation, this article touches upon some important implications for computers in the classroom. First, he's correct in suggesting that a computer could be used for problem solving and elementary "research". However, he seems to suggest that the kiddies use this shiny little monster as a gimmick for doing what Teach wants them to do. In other words, the teacher creates or cribs 49,000 problems and the student has to do 48,999 in order to get an "A", 48998 to get a "B", etc.

Several years ago, calculators were the great shiny, intrinsic motivators--alas, they have faded and have made planty of room for their big step-brothers, the computers.

A beginning acquaintance with computing is the most exciting thing that has happened to me in years, and I see it as a whole new philosophy of thinking rather than the science of the million dollar pencil!

Ron Edelman

Sternberg, Warren
Computing in the High School--Past, Present and Future--and its Unreasonable Effectiveness in the Teaching of Mathematics
AFIPS Journal, Vol. 40, Spring 1972
Pages 1051-1058

This article is concerned with two basic ideas, the use of the computer in the schools and why the computer is effective in teaching mathematics.

Computer use in schools is in four main areas: 1) Computer assisted Instruction "is a programmed text with the computer turning the pages" and is not widespread due to the high cost involved; 2) Student use of canned programs; 3) Student programming; 4) Computer courses, which have not been a part of the curriculum in many schools but are on the increase.

Computer usage is mainly limited to math classes, where the teachers need to know how to program and how to teach students programming and debugging techniques. In other subject areas, the teacher only needs to know how to call a canned program and input the necessary information. Materials and training tend to be more available to mathematics teachers; to help the non-math teachers, institutes and specialists must be made available for training.

The use of computers will have a definite impact on mathematics classes. The emphasis will shift from "what it is" to "what you do to find it". There is also a subtle difference in the computer attitude toward variables and the conventional mathematical attitude, and this difference seems to facilitate working with mathematical expressions.

Increased emphasis on the use of algorithms which occur naturally in programming and flowcharting, will improve problem solving techniques. Many math topics are algorithmic in nature. By using an algorithm approach, students should acquire a higher level of understanding of mathematical ideas. Flow charts lend themselves to this type of problem solving and students using flow charts seem to do better in their math classes.

Finally, the use of computers has a singular motivating power for many students. Students will often work hard on computer problems when they will study almost no other subject.

Fred Board

Twelker, Paul A.
Designing Simulation Systems
Education Technology Vol. 9, No. 10, October 1969
Pages 64-70

Teaching Research has developed a three-phase approach to designing instructional simulation systems. This approach involves (1) determining what to teach, (2) determining how best it might be taught, and (3) validating the system.

Thirteen specific steps are described in the article and an effort is made to expose the vital decision points in each. The steps are (1) define the instructional problem, (2) describe the operational educational system, (3) relate the operational systems to the problem, (4) specify objectives in behavioral terms. (5) Generate criterion measures, (6) determine appropriateness of simulation, (7) determine type of simulation required (interpersonal-ascendant simulation, machine or media-ascendant simulation, or nonsimulation games). (8) Develop specifications for simulation experience, (9) develop simulation system prototype, (10) try out prototype system, (11) modify the prototype system, (12) conduct field trial. (13) Make further modifications to the system deemed appropriate from field trial experience.

Included in the Step 6 outline are a list of situations in which simulation may be useful, and a list of arguments against simulation. A table for use in step 7 presents the relative advantages (in terms of 18 instructional factors) of each of three types of simulation techniques.

Lloyd Fraser

Vitelli, John L.
Computer and Television: A Joint Venture for Education
Educational Technology, Vol. 12, May 1972
Pages 44-47

There exist today a slow but growing acceptance of two new educational media, television and the computer. Television, however, has been used primarily to extend rather than improve instruction; while computers, with their enormous improvement potential have severe drawbacks in terms of costs-per-student. Merging the two tools may help to overcome the shortcomings of each.

Educators can use a number of methods to accomplish this. First, television monitors can be connected to a standard CRT (cathode ray tube) terminal. At present only about five students can view a terminal during a demonstration. By using monitors a class of 25 or 30 can view what is happening. Having the computer at his fingertips allows the instructor to explain or demonstrate concepts much more easily and rapidly. In teaching about the computer this is especially useful. Other advantages are that the CRT and monitors operate quietly as opposed to the clattering of a standard teletypewriter, and the CRT outputs data about three times as fast as a teletype.

Second, the CRT computer terminal can be used in the preparation of videotapes for classroom use. This is useful whenever the same presentation is to be given several times. The teacher can use the CRT device for viewing, then switch it to "line" and use it to run programs.

Third, the same CRT device can serve as either a television moniter or a computer terminal in a learning center. The student can view a videotape of some presentation and then either construct his own programs or use one of the prepared programs to run his experiment.

The equipment described is available at many institutions of higher learning and will soon be available at elementary and high schools. By using one or more of the three methods it is possible to reduce cost per pupil expenditures and at the same time provide a more rich educational environment.

David Dempster

CHAPTER V

THE COMPUTER AS A TEACHER

by

Marsha Dyer

Mike Dunlap*

Francis Farthing

David Moursund*

Eugene Thoms

*Mike Dunlap, University of Oregon, Eugene, Oregon, served as group leader for this chapter, and David Moursund, University of Oregon, served as editor.

SECTION A: OVERVIEW OF CAI

by David Moursund

Jane Doe, a fourth grader, sits down in front of a TV-like screen with a typewriter-like keyboard. She turns on the machine and types in her student identification number. The screen lights up and displays: "Hello Jane, it's nice to see you again so soon. You did a very good job on your arithmetic earlier this morning. I believe, however, you need more work on spelling. What would you like to work on now?" After a moment's thought Jane types in speling. The machine responds: "Good, I can see you certainly need some work here. Put on the earphones. When you hear a word on the earphones try to type it on the keyboard."

As the lesson proceeds Jane is presented with a sequence of words via earphone, and later a sequence of words via pictures of objects and of situations on the TV screen. The machine records her responses, corrects her errors and presents an appropriate level of difficulty of new words to be spelled. It uses information of Jane's spelling success on previous days, as well as standard spelling lists for fourth grade students.

Science fiction? No! Such CAI is possible today--and is available to a few students who happen to be going to schools located near the major centers of research and development in CAI.

A Very Brief History Computer assisted instruction involves an interaction between a learner and a computer. CAI is often classified into several categories. Drill and practice is much like computerized flash cards, and is commonly used for arithmetic, or vocabulary drill. In the tutorial mode of CAI the computer presents instructions in a manner similar to a programmed text. Intricate branching, based upon the student's responses, may occur. In the dialogue mode CAI is intended to be much like a one-to-one interaction between student and teacher. Considerably more progress in the field of computer understanding of natural languages (i.e., in artificial intelligence) is needed before much good dialogue mode CAI materials will exist.

All three forms of CAI just mentioned involve continual interaction (in "real time") between a learner and a computer. Before the advent of time-shared computing this meant that a computer could serve only one student at a time in a CAI mode. Thus relatively little significant progress occured in CAI during the 1950's.

During the early 1960's time-shared computing began to emerge, and several significant CAI projects developed. Pat Suppes' work (at Stanford University) is well known. He developed drill and practice materials in mathematics grades 1-6, CAI materials for teaching logic, language courses, etc. His materials were tested in a number of schools, and provided good insight into some of the capabilities, limitations, and problems in CAI.

In 1959 Donald Bitzer at the University of Illinois began the development of a keyboard TV-like terminal for use in CAI, and appropriate software for his system. This project has continued until the present; PLATO IV is perhaps the best and most sophisticated CAI system currently in existence.

As time-shared computing became available to more and more schools literally hundreds of small CAI projects developed. Dozens of CAI languages were developed and tried out. By and large most of these projects were small and had relatively little impact upon education. To illustrate, it is an almost "trivial" matter to write an arithmet drill and practice program in BASIC. Problems of an appropriate degree of difficulty are generated using a random number generator, and correct answers are computed by the computer. The student's response to a question is compared to the correct answer and a record is kept of the number of correct and incorrect responses. Feedback to the student is a reward (THAT IS CORRECT) or a punishment (NO, TRY AGAIN).

CAI Software and Hardware As previously indicated, the typical minimal facilities needed for CAI are computer access (perhaps via a single keyboard terminal) and CAI lessons. Thus CAI can be carried on at any time-shared computing facility. However, considerably more or better hardware and software are needed if one is to have an effective instructional system.

On the hardware side of things one needs a computer system with appropriate computer power and auxiliary storage, and with appropriate terminals. A model 33 TTY is an exceedingly minimal terminal facility. It is noisy and slow. It lacks both audio and video capabilities. The gas plasma display screen terminal being developed by Bitzer at the University of Illinois represents a "good" CAI terminal. Input by the student is via keyboard or by touching the display screen. Output to the student is via display screen, via colored or black and white slides projected on the same display screen, or via audio from a random access recording device. (The student can also record his answers onto this device, for later review by the teacher. This would be useful in languages courses.) Bitzer's CAI terminal is now in limited production at a cost of about $5000 per terminal.

On of the software side one needs a computer language suitable for writing CAI lessons, and a software system suitable for presenting lessons. Minimally, an interactive language such as BASIC can be used for both purposes. More desirable, however, are the languages which have been developed specifically for CAI usage. One of the first of these was COURSEWRITER I, developed by IBM. As previously mentioned, many dozens of these languages have been developed. In general CAI software consists of two parts. One part is a language for writing CAI materials; the other part is a system for presenting the lesson material to the student. Examples of such systems include COURSEWRITER II (IBM) and IDF (Hewlett-Packard).

Current National Status In recent years there has been a decrease in the total number of CAI projects receiving federal funding, and an increase in the funding levels of a few large projects. In particular, large amounts of National Science Foundation funding are going to the PLATO project (University of Illinois) and to TICCIT (University of Texas and Mitre Corporation). In both cases rather large systems are being field developed and implemented to test the economic feasibility and current technological feasibility of providing CAI to large numbers of students.

CAI is also beginning to make significant progress in areas of education which are typically quite expensive (medical school) and/or unsuccessful (inner-city, disadvantaged kids). Major projects are going on at several medical schools and in large cities such as Chicago, Los Angeles, and New York. By and large, however, CAI has not yet "arrived".

CAI in Oregon To date CAI has had almost no impact upon education in Oregon. No major federally funded projects have occured here, and relatively little research or development work has been done.

Of course, there are minor exceptions. At both Oregon State University and the University of Oregon a few small CAI projects have gone on, and a few students have used the materials. (OSU leads UO in this regard.) At the public school level the Suppes drill and practice materials in arithmetic grades 1-6 are available on the Multnomah County IED Hewlett-Packard system, and on the OTIS Hewlett-Packard system. The materials have received considerable use in the Portland area. Mike Dunlap, (formerly of Centennial High School, but now at the UO) experimented with the system for use with remedial secondary school students during 1971-72. The results were not encouraging.

The Future It seems clear that CAI will eventually have a major impact upon education. Cost studies at the PLATO and TICCIT projects suggest that good CAI systems must be quite large in order to be cost effective. Thus Bitzer speaks of 4000 terminal systems driven by a 4-CPU CDC 6400 computer system. Bitzer's current system has about 200 terminals, and this will be expanded by another 500 terminals during the next year. Such massive CAI systems will require careful planning and cooperation between various educational levels and organizations. They will require large amounts of money for their initial implementation. Thus, for the most part, it will be at least another 10 years before CAI begins to have a major impact upon public education.

The Literature CAI is a popular topic for research and writing. Consequently many hundreds of articles have been published. The beginner should probably seek a book of readings, or a series of "overview" articles. Two of these are listed below.

Atkinson, Richard and Wilson, H.A. (Editors) Computer-Assisted Instruction: A Book of Readings, Academic Press, 1969.

Daniels, Alan (Editor) Educational Yearbook 1971-72, The British Computer Socitty, 29 Portland Place, London. (This issue is devoted mainly to CAI)

Heimer, Ralph T. (Editor): Computer-Assisted Instruction and the Teaching of Mathematics, (Proceedings of a national CAI conference held at Pennsylvania State University.)
The National Council of Teachers of Mathematics, Inc., 1969.

SECTION B: COMPUTER ASSISTED INSTRUCTION: SOME CURRENT LITERATURE

by Francis Farthing

Although many tupes of machines and pieces of equipment have been introduced into the educational environment with the promise of significantly improving the quality of instruction; none have produced the significant improvement initially promised. These have included motion pictures, television, language laboratories, programmed instruction, and teaching machines.

Computer-assisted instruction (CAI) is a comparatively recent educational concept, having been on the educational scene for less than fifteen years. IBM researchers developed some of the first CAI courses in 1959. They were designed for teaching stenotyping and binary arithmetic (2).

The development of time-shared computing facilities during the early 1960's, and the rapid performance/price gains made possible by third generation computer hardware have helped to make CAI financially feasible. Many educators feel that CAI is now a powerful tool for improving the effectiveness of our school systems.

Not all educators, however, are convinced that CAI is really even a form of teaching. Some prefer to reserve the use of the word "teach" to applv "...only to advanced methods of training which permit a student to handle complicated problems of whose form and substance he be given a very broad outline."(6) Other educators see it only as an expensive tool for assisting students with a low-level learning process (drill and practice).

Just what is the potential for CAI and what is its present status? This is the concern of this paper. The author feels that a lack of knowledge regarding the developments that have been made to date and the potential developments that can be expected in the future have tended to make educators more skeptical than they should be. Certainly, the concept is exciting and worthy of all the attention one can give it.

Currently, the proponents of CAI have identified three fundamental characteristics of learning which they feel the computer is able to meet and which are essential to the sustained interest of the learner. The first is feedback. An effective CAI program responds immediately to the student's input, gives immediate feedback to the learner regarding the correctness of his answer, and supplies him with information regarding the nature of his errors.

The second characteristic for good learning that CAI can provide is an environment whereby the learner is actively involved. In CAI one typically has an interactive computer terminal, and the material being presented requires continual interaction on the part of the student.

A third desirable characteristic is the individualization of instruction. This is a strong point of good CAI materials. Lessons can be programmed for the specific level of achievement of the learner and for his specific interests and ability.

There is no disagreement among educators that these three characteristics are important, indeed, vital components of the teaching-learning process. There is also no argument that computers cannot be used effectively to satisfy all three of these demands for good teaching in the area of drill and practice. Currently, then, the main concerns with CAI are that it does not meet the needs of an "open educational setting" and that it is too costly.

Hall (2) lists the following four definitions of computer-assisted instruction*. The computer may be used:

---

*Editor's note: Hall, like many educators, confuses CAI with the entire field of use of computers in education. The current chapter is concerned only with his fourth definition.

(1) as a laboratory computing device. This type of program consists usually of a single terminal in a classroom, thus permitting students direct access to a computer. They can then develop programs pertinent to the course work they are taking. Hall estimates that currently about 500 high schools in the United States are using such an approach.

(2) as a record keeper and retriever. Here the computer is primarily used by administrators for batch processing data about students.

(3) in simulation. An important current example is training programs, to provide diagnostic-type experiences for physicians in training. Hall describes one such simulation device:

> Sim One is a lifelike device with a plastic skin which resembles that of a human being in color and texture. It has the configuration of a patient lying on an operating table, the left arm extended and ready for intravenous injection, right arm fitted with blood pressure cuff, and chest wall with a stethoscope taped over the approximate location of the heart. Sim One breathes; has a heart beat, pulse, and blood pressure (all synchronized); opens and closes his mouth, blinks his eyes, and responds to four intravenously administered drugs and two gases administered through a mask or tube. The physiological responses occur automatically as part of a computer program. (2)

(4) as a tutor. This appears to be the most common use associated with the term CAI. According to Hall:

> There are various forms which this definition might assume, the most common being that of providing drill and practice problems to students at a terminal and the most complex being that of sequential exposition which provides the primary source of instruction for the student. In this latter form, a relatively complete course is presented to the learner by means of a computer. (ibid)

Some authors consider tutorial systems and drill and practice systems as separate systems.(7) They categorize drill and practice systems as systems designed to help a learner in the maintenance and improvement of a skill and tutorial systems as systems designed to assist the learner in the acquisition of a skill. It appears that at the present time most CAI consists mainly of drill and practice type programs. Fitzgibbon states:

> Outside of using the computer to teach data processing practices, programming languages, and problem solving techniques, widespread use of CAI in public school classrooms has been limited to drill and practice programs in mathematics and English for elementary school level...CMI, where the computer monitors the instructional program but the student does not interact directly with the computer, is being developed and pursued in many widely scattered situations, but no broadly-used program is yet on the scene. Drill and practice techniques are producing evidence which supports the use of CAI for such activity and clearly indicates that programs such as these produce results worth the investments necessary to carry them off.
>
> Much work in the tutorial mode of CAI is being done at the college level and in military training institutions where more sophisticated and costly computer equipment is available. Problem-solving, gaming, and simulation techniques are exciting but again, part of a costly hardware system.(1)

Although the jury is not yet in regarding the use of computers for assisting in instruction, evidence is available that attests to the effectiveness of drill and practice CAI programs. Rosenbaum says that, "enough evidence now exists to convince all but the most skeptical that the administration and supervision of drill and practice activities in the classroom can be facilitated and indeed substantially improved through use of a well-designed CAI (computer-assisted-instruction) system." (5)

Vinsonhaler and Bass (7) have summarized the results of ten independent studies of CAI programs involving over 30 separate experiments with about 10,000 subjects within the content areas of language arts and mathematics. Each of the ten programs studied used a mode of CAI described as drill and practice. Most of the studies compared an experimental group that received traditional instruction augmented with CAI with a control group that received traditional instruction only. Some, however, controlled for possible Hawthorne effect by permitting some CAI experience for control subjects. All evaluations were norm referenced in that all the studies concerned methods for improving the performance of students as measured by standardized tests.

There were three major language arts programs included in the summary. They were from California (East Palo Alto), Michigan (Waterford Indicom), and New York (Harcourt Brace). The California study included 88 subjects in the first grade. The Michigan study included 68 subjects in the fourth grade while the New York study included 1800 students from a national sampling of fourth, fifth, and sixth graders. The results in all instances favored the CAI groups who made from one tenth to four tenths of a school year's growth more than the control groups.

Seven mathematics programs were studied. Two from California studied, respectively, 182 pupils in grades 4-6 and 665 pupils in grades 1-6. Two programs from New York studied 6534 pupils in grades 2-6 (3000 and 3534 respectively). Two studies of the Waterford program were reported. One, of the 1968-69 program, reported regarding 391 pupils in grades 3-6. The seoond reported on the 1969-70 program regarding 335 pupils in grades 2-6. One program from Mississippi studied 515 pupils in grades 1-6.

Except for two major exceptions (explained by lack of time in the program and by testing instruments) all these studies significantly favored the CAI groups. Regarding the effectiveness of CAI the authors conclude:

> The effectiveness of CAI over traditional instruction seems to be a reasonably well-established fact in drill and practice for both mathematics and language arts, when performance is measured by SAT and MAT type tests. In the field of elementary education, there appears to be little reason to doubt that CAI plus traditional classroom instruction is usually more effective than traditional instruction alone in developing skills--at least during the first year or two. What remains in doubt is the advantage of CAI over other, less expensive methods of augmenting traditional instruction and the long-term effects of CAI. There are indications that the effects obtained with CAI might be obtained through less expensive means. For example, one of the studies reported by Suppes and Morningstar (1969) suggests that an additional 30 minutes of ordinary classroom drill

> and practice can accomplish the same results as a 15-minute CAI program. Another possibility is programmed instruction. Studies have shown that CAI may actually prove inferior to programmed instruction under certain circumstances. For example, Dick and Latta (1969) compared the performance of eighth grade students on CAI and PI presentations of mathematics concepts. The results favored PI over CAI on all measures.
>
> In the field of drill and practice for mathematics and language arts, what we now need are studies which compare CAI with other nontraditional methods of instruction and which attempt to identify the underlying bases for the CAI effects. With regard to the latter, we presently do not even know the major sources of the advantage of CAI over traditional instruction. The advantage could be due to direct effects of CAI instruction; to "novelty" effects which decline over a period of years; to changes induced by CAI in behavior (additional classroom drill); or to changes in student behavior (voluntary additional practice). Research studies have established that CAI is effective; we must now consider the more sophisticated question--"How does CAI improve instruction?" (7)

A word of caution, however, may be in order regarding the aforementioned evaluative studies which were all based on standardized test results. Hall comments regarding the difficulty of assessing the effectiveness of CAI programs:

> One of the difficulties in evaluating CAI today is the choice of an adequate criterion. When instruction is individualized and is adapted to the specific weaknesses and strengths of the pupils, it is inappropriate to use a norm-referenced measure for evaluating student progress. A more appropriate apprach is to utilize a criterion-referenced measure to determine when the students have reached the desired level of achievement. The difference in objectives and evaluative devices makes it difficult to compare progress in CAI with progress in the conventional classroom. Even with these problems of evaluation, a consistent result in the use of computer-assisted instruction has been that the same amount of material has been learned in a CAI environment as in a conventional classroom, although with a considerable saving of time in favor of CAI. (2)

Many skeptics of CAI have expressed concern that with such programs students feel depersonalized. Adequate empirical data is not presently avaliable regarding the attitude of pupils working in CAI programs. According to Hall, preliminary evidence indicates that students involved in

CAI programs do not feel depersonalized.

It thus appears that CAI is a valuable concept even though its current main use is for drill and practice type activities in the language arts and mathematics areas of the curriculum. Risken, in discussing the use of computers in teaching composition, pretty well summarizes the limitations of current CAI programs:

> Even in computer-aided or computer-based instruction facilities, the most that can presently be done is to examine a student's answer and check for the presence or absence of particular words or their synonyms. And while fairly efficient syntactic parsers are being developed, there will still be a delay of at least a few years before computers can "talk" economically and with any freedom with humans...a completely computer-based course in effective writing then, is not currently feasible.(4)

It would seem then, that the future of CAI hinges on whether or not it can be adapted to fit the needs of children in an "open" educational environment. In such an environment the main philosophical concern is to give the learner as much control of his own learning experience as possible. Resnick states:

> Those individuals who will be in the best position to control their own learning experiences are those who command the greatest range and depth of "learning skills." The more that individuals can organize bodies of knowledge, search texts or other presentations for useful information, and analyze new skills in order to "program" their own acquisition sequences, the more they will be able to learn independently of organized programs and skilled teachers. Thus, with respect to choosing educational objectives on which major technological resources will be concentrated, the fundamental principle suggested is to give preference to those objectives that are "generative"--i.e., that give the learner some increment of power in learning something else (cf., Bruner, 1964). (3)

Thus, it seems that the future of CAI depends on whether or not the computer can be utilized to assist the learner in becoming more independent and capable of generating his own learning experience.

Bibliography

1. Fitzgibbon, Norinne, et al. "CAI: Where Do We Go From Here?" Elementary English, November, 1970, pp. 917-21.

Bibliography, Continued

2. Hall, Keith A. "Computer-Assisted Instruction: Problems and Performance." Phi Delta Kappan, June, 1971, pp. 628-631.

3. Resnick, Lauren B. "Open Education: Some Tasks for Technology." Educational Technology, January, 1972, pp. 70-76

4. Risken, John. "Written Composition and the Computer." Educational Technology, June, 1972, pp. 46-51.

5. Rosenbaum, Peter. "Toward the Automated Workbook." Educational Technology, August, 1971, pp. 36-38

6. Stehling, Kurt. Computers and You. World Publishing: New York, 1972, pp. 308.

7. Vinsonhaler, John, and Bass, Ronald K. "A Summary of Ten Major Studies on CAI Drill and Practice." Educational Technology, July, 1972, pp. 29-32.

SECTION C: THREE COMPUTER ASSISTED INSTRUCTION SYSTEMS

by Eugene Thoms

The Index to Computer Assisted Instruction (3) lists 910 CAI programs; of those, 434 have been designed for use at the secondary school level and 588 at the college or adult level (many are intended for both areas of instruction). Of course the majority (nearly 60%) of the programs fall into the math science category. The remainder cover a wide variety of subject matter areas. The programs listed range in length from a few lessons to full semester courses.

Rather than attempt to review a number of individual programs, this section will discuss the features of three major CAI systems which are either currently being tested or are to be tested in the near future. The development of each of these systems has been funded to some extent by the National Science Foundation. These systems are representative of CAI today and from all appearances will certainly have an effect on future education.

THE NEWBASIC SYSTEM(1)

CAI work in most situations is restricted by the system the authors have to work with. This program tries to overcome some of the drawbacks, to allow individual instructors and students some control over the CAI they receive. The project is currently being tested in Pittsburgh. The experiment is being tested in secondary schools on a regional basis.

The author Dwyer,(1) suggests that the use of CAI is helpful in getting students into writing their own CAI materials. He cites the example of a student who, after using CAI programs, become interested in their construction. The construction was explained and he then proceeded to write a drill program for a German class. At first it was simply a "paired-associative" drill; but soon other students became interested and German language comments to correct and incorrect responses were added to the drill program. Since it's difficult for a person not to want to know what comment has been made about him, the result was that the students using the new drill learned more German than was required by the original drill.

Part of the purpose of the NEWBASIC project was to gain insight into the requirements for a good CAI system. Initial conclusions as to the required features of a good CAI system were the following three:

1) The first is easy system access in all modes.
2) The second requirement is that full power of general purpose computing should be available to all users.
3) The third requirement is that the system should be "approachable" on the educator's terms. (1)

By easy system access, it is meant that the language (NEWBASIC in this case) should be self-prompting and uncomplicated. Easy access allows students to concentrate on the content of the CAI rather than the communication with the system.

By having the full power of the system available, the staff and students can construct or modify CAI programs; this helps to make CAI more dynamic and also gives better insight into the capabilities of computers.

The third requirement of the system can be thought of as not requiring users of the system to be qualified computer scientists. A good system should be usable by faculty (and students) who have had little or no training in the computer field.

It is possible to start from scratch and develop a language suitable for CAI. Another approach is to expand and/or modify an existing language. The language NEWBASIC (NBS) is an expansion of BASIC.

For those who are familiar with BASIC or are interested, standard BASIC for this system has been expanded in the following areas:

1) Extension as an algorithmic language: NBS allows such things as multiple data types (REAL, INTEGER, COMPLEX, DOUBLE REAL, etc.), suffix control, (FOR, WHILE, UNTIL, ELSE, etc.),

multiple statements (IF X=5 PR. "LARGE" ELSE PR. "SMALL", GOSUB 120), picture formats, dynamic string allocation, and extensive file handling commands.

2) Extension as a CAI language: string functions that allow for the scan of user input are provided to allow greater freedom in responding to questions posed by drill or tutorial programs.

3) Extension as an interactive language: a) NBS has the interactive capability of interpretetive languages such as JOSS. Program execution can be halted, and variables examined, experimented with, or changed, using direct mode. Execution can be resumed at any point in the program with a direct "GO TO" command. b) The @NBS feature permits a second level of NBS access for students who are under control of a first level NBS CAI program. c) Executive commands (e.g. looking at directories or contents of files, or asking for the editor) can be issued during either the execute or create phases of NBS.

4) Extension to meet new needs: users can write functions to suit their own particular needs, using assemble language, FORTRAN, or NEWBASIC. These can be added to the list of public library functions any time that general interest warrants. There are over 300 library functions at the present time. (1)

Problems with implementing loops are avoided by a new statement called PASS which acts as a loop. The invocation of PASS automatically sets up counters and incrementing.

NBS has the ability to scan free form responses for certain combinations of key words. This feature allows the student more freedom of expression.

Among the convenience features of NBS is a form employed to aid in writing CAI, especially at a beginning level, This form can be used for each discrete area of a tutorial. Each area is then broken down into 3 sections: presentation and scan, expected response, and expected error. The form shows the interconnections of those areas and gives the range of line numbers. This range is helpful in avoiding clashes when the various areas are lumped together.

In the preliminary report, the schools have experienced an increase from 25 to 100 connect-hours per month in the change from FORTRAN to NBS. A number of CAI programs have been completed, and are now being tested. The only economic information available is that the test school system would need to increase it's budget 2% to support 200 terminals for grades 8-12.

THE TICCIT SYSTEM(2)

TICCIT (time-shared interactive, computer controlled information television) is being developed bv Mitre Corporation in conjunction with the University of Texas and Brigham Young University. The goal of this system is that of supplying effective and economical CAI. This project, like PLATO, is receiving a large amount of National Science Foundation funding.

The input-output device for this system will consist of a color television, keyboard and headphones. The television set will be capable of displaying textual material in a type size suitable for easy reading. Both text and background color may be specified. Audio information is stored on random access record players. Besides displaying text and graphics, the television can also be used for play-back of standard video tapes. The use of television sets as a major part of a terminal is quite important. It suggests the possibility of putting CAI into the general public's homes via cable TV.

This system is composed of two small computers, one to provide terminal service and the other for standard computer operations. Program material will be kept on disk storage. Console connections will be either direct or via cable television lines. Each terminal will have an integral curcuit memory device to alleviate the problem of "refreshing" the television screen display.

the cost of such a device is a major expense for the console; but the prices in the area of electronics are being reduced rapidly. (In 1969 Mitre Corporation did a feasibility study of a 10,000 terminal CAI system based upon using modified TV sets as terminals. Estimated costs were under 50¢ per terminal hour.)

THE PLATO IV SYSTEM (4)

The PLATO IV project at the University of Illinois is perhaps the best representative of what CAI can be. The PLATO (Programmed Logic for Automatic Teaching Operations) project began in 1959 under the direction of Donald Bitzer. In its early years it received support from Control Data Corporation in terms of a free computer. One of its first CAI programs (1960) was a series of lessons designed to teach computer programming.

The PLATO IV project is an outgrowth and logical continuation of the original project. Over the years the PLATO system has been used by large numbers of students in a wide variety of courses. Statistics have been gathered on the experience of 100,000 student hours of usage of the system. These statistics provide the basis for design of newer and better systems. They also provide the basis for projections of the costs per student hour of usage for the newer, larger systems.

The current (PLATO IV) project is operating under a $5 million grant from the National Science Foundation along with a large amount of funding from the state of Illinois. The hardware and software to support a 1000 terminal system is being developed. About 200 terminals have already been put on the system, and another 500 terminals are now being manufactured.

As envisioned by Donald Bitzer, a single PLATO system will ultimately be able to handle 4000 terminals. The computer to do this will be a Control Data Corporation 6400 with four central processing units. Communications costs will be minimized by multiplexing on leased lines and by use of educational television channels. (One TV cable channel can accomodate one thousand remote CAI terminals.)

The present test version of PLATO uses Bitzer's terminals, which cost around $5,000 each. Mass production should reduce this cost considerably, Bitzer's projection is about $1,800 each. Terminal features include a plasma display screen, a random-access audio device, a random-access slide projector, and a sensor system in the display which responds to touch. This type of display does not need to be restimulated as is the case for television.

The estimated cost per student hour when PLATO is fully operational is 50 to 80 cents (projected 1980 costs based on 1970 dollars). A major consideration in the projected costs involve software (instructional materials). If widespread use of the software is realized, their production costs would be minimal. The computer itself costs in the neighborhood of 5 million dollars. It has much faster memory access than presently possible on memory discs. It gives terminal response in a tenth of a second and allows each user nearly complete computer computational power.

The state of Illinois has shown considerable interest in this system and is supplying about half the demonstration cost. One reason for this is because it has been predicted that eventually such a system could help the state accommodate a 20% school enrollment increase at a 2% increase in their budget for education.

FINAL REMARKS

CAI is on the verge of being ready to make a significant impact upon the world of education. The two major projects, TICCIT and PLATO, represent two major ideas on what might eventually come to be. TICCIT is testing ideas which would allow the home TV set, and community antenna cable TV lines, to be hooked into computers to bring CAI into millions of homes. PLATO is testing an advanced CAI terminal and system designed to provide economical CAI in units of 1000 terminals. The University of Illinois itself, for example, may eventaully have 1000-2000 terminals on its campus. This is bound to have a significant impact upon its educational program.

A fundamental question about CAI is the effectiveness, or the cost/effectiveness of such systems. Considerable research remains to be done to measure the effectiveness of CAI against other modes of computer hardware along with economics of scale (4000 terminal systems) suggest that 40¢-80¢ per student hour of usage may well be possible. If so, we can expect CAI to become commonplace in the home and the school during the next 20 years.

Bibliography

1. Dwyer, T. A. "Teacher/Student Authored CAI Using the NEWBASIC System" Communications of the ACM, January, 1972, pp. 21-28.

2. Hammond, Allen L. "Computer Assisted Instruction: Two Major Demonstrations". Science, June 9, 1972, pp. 1110-1112.

3. Lekan, Helen A. Index to Computer Assisted Instruction, 2nd ed., Sterling Institute, Boston, Mass., January 1970.

4. Levine, Roger E., et al. The Emerging Technology. McGraw-Hill Book Company, New York, 1972.

5. Alpert, D. and Bitzer, D. "Advances in Computer-Based Education." Science, March 20, 1970, pp. 1582-1590.

Section D: Abstracts of related articles for Chapter V

Abelson, Philip H.
The Fourth Revolution
Science, Volume 177, (July 14, 1972), page 121

This editorial by the editor of Science proposes that much of the research and development in education be directed into the investigation of CAI as a medium offering better quality teaching for less money in some subjects.

Abelson quotes a report by the Carnegie Commission on Higher Education which predicts general use of computers in libraries by 1980 and in schools by 1990. Although the report cites four major advantages of CAI: 1) independent study, 2) richer variety of courses and methods of teaching, 3) infinite tolerance and patience, and 4) off-campus instruction, the authors predict a long delay to its acceptance due to the "inertia of higher education" and the disenchantment with CAI so far.

Abelson proposes that a larger project is needed to successfully demonstrate cost- and quality-effectiveness of CAI. He mentions two systems now under development; one of them, the Plato system at the University of Illinois, is scheduled to begin in the fall of 1973 a two-year demonstration of new hardware and new CAI programs on 500 to 1000 terminals.

He ends the article with a call for "critical and imaginative thinking" about the possible adverse effects of the new technology.

Marlyn Kern

Calingaert, Peter
Teacher Control in Computer Assisted Instruction
1 April, 1972, 23p. (Available through ERIC)

Systems of computer-assisted instruction (CAI) can be classified according to whether the author, student, or teacher controls the interaction between the student and the computer. Both author-controlled and student-controlled CAI have the advantages of individualized instruction, privacy for mistakes, and flexibility, but are tremendously expensive. Student-controlled CAI further allows a student to be much more active, but also makes supervision difficult. A teacher-controlled system used as another teaching tool before a whole class of students is much cheaper than the other types of CAI, and adds to the computer program an intelligent subsystem, the teacher, to filter input and modify the stream of presentation. Teacher-controlled CAI gives up the advantages of individual attention, privacy and flexibility, but the criterion of cost-effectiveness makes it an attractive possibility in the hands of a skillful teacher.

Thomas Stone

Hammel, David G.
An Introduction to Computer-Assisted Instruction
Modern Data Vol. 5, No. 10, October 1972
Pages 42-46

Generally CAI refers to individual tutoring using a computer-controlled presentation of material. Multiple students working simultaneously at terminals and interacting independently with the computer make up the system. Ideally a student should be able to study any lesson in the CAI repertory if he has the qualifications. She should be able to progress at her own rate independent of time and instructor availability, being required to demonstrate an understanding of the current subject matter before being faced with new concepts. The computer maintains records of student progress in order to tailor the lesson to his needs and also to reveal weaknesses in the lesson (providing for modification of the system). At any point the student should be able to ask for clarification of material. When the answer is inadequate, assistance from a proctor should be provided.

The main argument for CAI is that it provides a means for individualizing instruction as opposed to Tradition-Administered-Instruction (TAI). Typically, the TAI instructor aims his presentation at students with varied levels of aptitude and achievement. Individual needs of the students are not met and the slower students become frustrated while the brighter students lose interest. Unless the class is quite small, the TAI instructor is often not able to determine whether or not each student understands the current material. False concepts assimilated during a classroom lecture or failure to grasp the correct concepts may mean that the future material will not be learned properly.

CAI requires a blending of several disciplines. The basic lesson text should be specified by subject matter experts. Educational psychologists next introduce appropriate teaching strategies. Programmers then translate the lessons into software. Finally the system must be field-tested and modifications implemented into the system. Ideally CAI systems hardware must be developed along with the development of a CAI language with support software. Teletype equipment, currently used for communication because of its low cost, is largely inadequate.

Thus far, results obtained using CAI have not been significantly different from those using TAI. Matters are further complicated by costs much greater than TAI costs. This is attributed largely to a lack of hardware capable of dealing with student/system interaction adequately. Ultimately CAI will become competitive, with the establishment of a computer facility which will service thousands of terminals in schools, homes and industry. Costs will become tolerable through large scale time-sharing of its facilities. Industry could design and develop such a system within 5 years if adequate funding were available.

Frederic Daniels

Hammond, Allen L.
Computer-Assisted Instruction: Many Efforts, Mixed Results
Science, Vol. 176, 2June72, pps. 1005-1006

Computer Assisted Instruction (CAI) is generally seen as having a potential for major impact on individualized instruction. Resistance to change by school systems and high costs are presently the major obstacles in the path of this predicted impact. Also, CAI systems based on older computing systems have definite disadvantages. New CAI systems are being developed specifically for education.

M.I.T.'s Seymour Papert and colleagues have developed a mechanical "turtle" which responds to programs written by children and leaves a trace of its path. English based LOGO language is used. In trying to cause specific traces, the student learns various abstract concepts informally when writing the programs. This is an example of CAI for enrichment rather than the normal drill or tutorial generally expected of CAI.

Stanford University's Patrick Suppes and colleagues' drill and practice system for grades 1-6 has been rather extensively used and tested. These programs are used to assist teachers in arithmetic drill 5-10 minutes each day. The programs consider the psychology of learning when setting up the student's questions and problems. Testing has shown such a program can increase learning, although minimally more than good drill by a teacher.

At the college level, Stanford had good results with a Russian CAI course. This course produced more student learning, increased enrollment, and had fewer dropouts. The course was three times as expensive as normal, so it was discontinued.

Dartmouth provides nearly 100 terminals in various locations on campus. An estimated 90% of their 3000 undergraduates use the system, which has data available for social sciences and business, games, simulations, and normal computational features. BASIC is used on a system capable of handling 160 users with system failure averaging once per day.

A computer controlled microfiche system is being developed to give vocational-technical training for the Air Force. F.C. Frick and others of M.I.T.'s Lincoln Laboratory are developing the system that uses microfiche containing recorded directions and comments.

Scepticism is still prevalent despite the efforts in CAI to date. Second generation CAI systems should be capable of overcoming many of the present drawbacks.

Eugene Thomas

Hammond, Allen L.
Computer-Assisted Instruction: Two Major Demonstrations
Science, Vol 176, n9, (June72,) pp. 1110-1112

In the near future, two major CAI systems will be demonstrated for a two year period. Nine million dollars in support is being provided by the National Science Foundation. Each system will be evaluated by Princeton's Educational Testing Service under a separate million dollar contract.

The original PLATO (programmed logic for automatic teaching operations) system start occurred in 1960 at the University of Illinois and has already provided universities, colleges, and other schools with more than 100,000 CAI instruction hours. This system will begin its demonstration in the fall of 1973.

The PLATO system is based on a Control Data Corporation 6400 computer with four central processors and is ultimately capable of handling 4000 student terminals within a 800 mile radius. The demonstration will have between 500 and 1000 terminals, however. The CDC computer has .1 second terminal response time and allows extremely rapid access to blocked information since no slow disc memory is used for lessons.

The terminals for PLATO are perhaps its most sophisticated feature. It has a plasma display screen, keyboard, random-access slide projector, random-access sudio system, and touch sensitive display. The terminal can be connected by normal telephone connections, or for remote areas, as many as 1000 terminals may be connected by a single educational T.V. channel. The terminals can display up to 180 characters per second.

TUTOR is based on English syntax and grammer and is an easy programming language to use for teachers unfamiliar with computers. This author language has been extensively tested previously and should be reliable. TUTOR has facilities for judging student answers, dialogue preparation, generating graphics, high speed vocabulary search methods, etc.

The goal of the PLATO system is to increase the productivity of teachers and the process of education. The system is intended to be used in all areas of education with varying educational strategies. This system has and will continue to rely for the most part on materials produced by individual teachers.

A cost of 50-80 cents per hour per student is the forecast when the system is fully operational. This includes development of course materials and the time beyond the normal 40 hour instructional week being sold to other users. Mass production of the terminals should also reduce their present $5,000 cost. The state of Illinois is providing nearly half the upcoming demonstration cost, since it has been predicted that such a system could allow an enrollment increase of 20% at an increase of 2% in the educational budget.

Ticcit (time-shared interactive, computer controlled information television) is being developed by Mitre Corporation in conjunction with the University of Texas and Brigham Young University. Its goal is that of showing that CAI can be produced and packaged in a way that is economical and effective.

Ticcit is not centralized. It is built around two small computers capable of handling 100 terminals. One will service the terminals, and the other will process answers.

Each terminal is composed of a color television display, headphones, and keyboard. Random-access computer controlled record players provide audio output. Terminal connections will be either direct or by cable television channels. The display can exhibit up to 17 lines of 43 characters with both the display and background color specified. Video taped materials can also be displayed.

The program materials will be developed by subject area specialists, programmers, and educational psychologists. This method would place the teacher in the role of diagnostician. The programs are designed to give each student a variety of approaches to learning, including touches of fun and humor.

This system will start its demonstration period in 1974 at the community college level. Standard and remedial courses in English and mathematics will be the initial program materials available. They hope to show a displacement of 20% of the teaching load.

The estimated cost of the system with terminals is $400,000. The cost per student per hour is estimated to be 35 cents, but this does not include the cost of producing program material.

The costs involved in computing are rapidly being reduced and the Ticcit and PLATO systems will likely have a major influence on education, especially if they prove to be effective and economical.

Eugene Thomas

Moorer, James Anderson
Music and Computer Composition
Communications of the ACM, V. 14, n 2 (Feb 1972)

Considerable work has been done using the computer as a musical performer and composer. An early ILLIAC experiment produced a string quartet that was composed entirely by computer. However, little work has been done on the problem of simulating human composition using heuristic (self-taught) techniques as opposed to random number generation. This article explained some difficulties involved, provided background, and discussed schemes already tried.

Problems:

1. Style - There are more styles than composers because a composer, as he matures, may have many styles, e.g., Beethoven. Modern music employs and accepts multiple combinations of sounds. This whole problem involves racial and cultural variances from Stone Age man to the ancient Greek to the modern American or African. Because of some fundamental differences--e.g., music that is atonal or aperiodic--some styles are omitted from the discussion.

2. Fundamentals - Rhythm is fundamental because man has always been confronted with periodicity: the hours, minutes, seconds, days, months, etc. A second fundamental is pitch, high usually indicating tension. It is related to human noises, the lower and softer often signifying lack of tension. Finally, sound perception mechanisms must be examined to discover why certain timbres and combinations of notes are pleasing to our ears while others are not.

As in speech, the musician may produce an infinite number of musical "utterances" or melodies. Many of these may be produced by formula, i.e., a sonata is an exposition, a development, and a recapitulation. Musical composition then, is in many respects, similar to a grammar that will cover a set of utterances. Music, however, goes beyond the analogy of grammar because of its effect--that which is conveyed.

Another question is that of the secual nature of music. The author uses analogy of building tension in music by increasing sound and raising pitch to the point of climax.

All of this to say, stated Moorer, that if we are to create computer music that appeals to an audience, it must be somewhat like music to which they are accustomed. The random recall of parts of previously heard music gives a statistical character to the music. People expect certain chords after other chords. This is one of the reasons for the success of the Markov methods of producing music by computer.

The most publicized piece of computer music is the ILLIAC Suite (described by Heller and Issacson) whose rhythms were chosen at random with the application of some rules of classical harmony for a string quartet. The melodies were constructed by the use of a Markov method.

The result is modern, the melodies often are unpredictable, chord sequences do not repeat, and some dissonance is used.

In another experiment, the pieces were composed in an orderly, sequential manner, the overall structure was determined first, and many parameters governing the composition were set by the operator. The rhythms were chosen at random. Then the melody notes were chosen carefully (occupying three times the space of the rest of the program). This process eliminates boredom caused by such things as long scalar runs.

Finally, the author presents five short single-voice examples without the related chord structures. He says that all are definitely "legal popular melodies." Computer time to produce them was less than ten seconds, which indicates the shallow nature of the generating heuristics. The author hopes that his thinking and his experiment will serve as a starting point for others.

H. Snelgrove

Suydam, Marilyn N.
Teachers, Pupils, and Computer Assisted Instruction
Mathematics Teacher, Vol, 16, No. 3 (March 1969)
Pages 173-176

Elementary mathematics teachers will soon be making decisions regarding instruction using computers, or computer-assisted instruction. There are several ways in which the computer helps the teacher with instructional tasks:

1. Tutorial - the computer-teacher initiates the question, for which answers are stored.
2. Drill and practice
3. Inquiry - students initiate questions asking for stored information.
4. Calculator - aid in computation.
5. Management - computer stores and analyzes data, helping teachers decide what to "prescribe" next.

CAI represents a real means of individualizing instruction, the "carrot" that has been dangling in front of the teacher's nose ever since classes were formed. True individualization depends on careful analysis of the mathematical content and the ways in which children learn.

The computer can help the teacher teach on a readiness level for each child. Programs can be written so that the faster learner can move very quickly, while the slower learner is given as much re-teaching as necessary. The computer is very patient at restructuring and repeating--provided patience is programmed in. CAI is only as good as the programmer. Few programs are yet available which meet the full potential of CAI.

Obviously CAI is unable to do the whole teaching job. There is still a need for interaction among learners, and between learners and teachers. Teachers will find that their classrooms won't be run by the computer--but the computer can be very helpful to the learning process.

Larry Nesland

Jerman, Max
The Use of Computers to Individualize Instruction
Mathematics Teacher, VLXV, No. 5, (May 1972)
Pages 395; 466-471

There seems to be some disagreement among definitions of "individualized", but proponents of each definition of individualizing instruction hold in common certain assumptions about education. These assumptions, which lead us to place as much hope as we do in Computer Assisted Instruction, are the following:

(1) Human beings are capable of learning much more than they are being asked to learn.
(2) There are many approaches to instruction that can be tailored to individuals.
(3) Careful diagnosis can bring insights into how one should individualize instruction.
(4) Reinforcement and feedback should be immediate during instruction.
(5) In each case the quality of education each person receives should be the highest possible.

One must know what it is that he wants to accomplish (objectives), how much money he has to spend, and what CAI systems and programs are available before he can make a decision about how to go about using CAI to assist in individualizing instruction.

Most CAI systems now in use are either quite large (100 or more instructional stations) or small (1 to 32 stations). Arguments for the large system are (1) availability to many users, (2) cost effective instruction, (3) more powerful CPU. Arguments for the small system are (1) low initial cost, (2) lower communications cost, (3) lower maintenance and installation costs (4) compatability with a larger number of schools that are willing to share programs and (5) most if not all commercially produced CAI programs available to large systems are available to the small systems.

In view of the number of years that the various developmental programs have been in operation, there are relatively few CAI programs currently available. The perfect system for individualization of instruction is far from being built and perhaps never will be realized. Some of the most persistent problems plaguing CAI are the following:

(1) Overselling educators to the potential of CAI.
(2) Relating CAI work to other classroom activities.
(3) Teacher acceptance.
(4) Integrating CAI with the rest of the curriculum.

The future of CAI is very promising as a means for individualizing instruction and probably the major problem to be faced is one of helping teachers to prepare themselves to use it. In order to use CAI with maximum effectiveness, a teacher should be knowledgeable in mathematics, structure of curriculum, psychology of learning and individual differences, and CAI instructional systems and programs.

Larry Nesland

Seltzer, Robert A.
Computer-Assisted Instruction--What It Can and Cannot Do
American Psychologist, v26, n4, (April 1971), pp. 373-377

The debate about the role of a computer in the classroom has ranged from the complete replacement of the teacher to the computer's inability to be more than a programmed textbook. What the computer can and cannot do is a matter of research and fact. What it should and should not do is based on value judgements.

A computer could be used for 1) drill and practice, 2) tutorial practice, to take over responsibility for development of a skill using a given concept, 3) dialogue-true interaction with appropriate response, 4) simulation and gaming, 5) retrieval and reorganization of information, 6) problem solving with computation and display tools, and 7) artistic design and composition.

The criterion statements with which to judge the computer's place in instruction are (a) If the computer poses a unique solution to an important problem in the instructional process, then it should be used regardless of the cost involved. (b) If the computer is more efficient or effective and the cost of its use to instruct is minimal, then it should be used. And conversely, (c) if the cost of development and use of the computer in instruction is relatively high with the relative efficiency or effectiveness only marginal, then the computer should not be used in the instructional process.

Observing the seven uses for the computer in terms of the above criterion at this time we find:

Drill and practice - In light of (c) a non-recommended use.

Tutorial practice - In light of (b) a valid use since the computer seems to be more effective in presenting blocks of information.

Dialogue - At this time the computer is unable to do this.

Simulation and gaming - A very good use. No other system can approach the richness of simulations done on a computer.

Retrieval and reorganization of information - This is justified by (b) above.

Problem solving - To use the computer is proper because no other system can perform calculations as quickly or as accurately.

Artistic design - Not at this time due to (c).

Larry Spencer

Umans, S.
Computers in the Classroom
Parent's Magazine, Vol. 46, (March 1971), pp. 62-63

Approximately 8,000 elementary school children in New York City are using computerized instruction for drill and practice work in mathematics. They use typewriter terminals (which are hooked to a central computer by ordinary telephone lines) to interact with the computer. Their lesson is printed by the typewriter (as directed by the computer's program).

The practive and drill sessions provide a combination of teaching and testing. If a student misses the same question after two attempts, the computer gives him the answer and then asks the question again to make sure the answer has registered. An average student is thus paced at this step-by-step mode and is not allowed to go on to more difficult examples until he proves that he understands the simpler material. Another important feature is that the computer automatically analyzes each student's performance and skips sections of the program which the youngster has already mastered. The computer also scores and times the lesson and prints out this information. The fact that the student can keep this print-out sheet is often a source of pride. It also serves as motivation for reviewing the lesson.

Most children are nervous at their first encounter with computerized instruction. However, they relax when they find out that the computer will not go on until they have answered; thus, they will not be "left behind". A few students become frustrated, but most like the feeling of independence.

Teachers' opinions of computerized drill and practice instruction are mixed. But most agree that it does help to individualize instruction because it frees them to devote more time to free-ranging discussion; this discussion reinforces the discovery approach for learning new concepts.

Programs more sophisticated than the drill and practice lessons are now being developed. These programs will hopefully stimulate their student user to think more critically and to answer open-ended questions with phrases or sentences. The computer will pick key words and certain combinations of words in the student's reply, and react to them by typing whole thoughts and leading questions.

Mathematics is not the only subject for which computerized instruction has been developed. Instructional objectives in foreign languages and consumer education are also being met. Another very important area for computerized instruction is that of remedial reading. The novelty and privacy of such instruction is often enough to induce a reluctant junior-high student to try to improve this most basic academic skill.

Don Weaver

Hicks, Bruce L.
Will the Computer Kill Education?
Educational Digest (September 1970)
Pages 10-12

This article discusses the advantages and dangers of Computer-Assisted Instruction (C.A.I.). One advantage of CAI is that a large variety of educational activities can be made available to the students. These programs can provide step by step instruction, which leads the learner through the material; or they can utilize the inquiry technique to gain insight into varied subject matter. Secondly, experiments can be performed on scientific equipment or they can be simulated without such equipment through the use of CAI. Thirdly, the student and teacher can both receive instant feedback on performance levels. This gives the teacher a tool to plan teaching strategies for each individual student. Finally, CAI can create a new environment for both students and teachers, since the computer can accept any or all possible solutions to a problem. It can tolerate the student's mistakes and encourage him to correct these mistakes. It must be realized that for the computer to be a patient and humane "teacher", the CAI programmer and lesson writers must be able to incorporate these qualities into their programs.

An obvious danger of CAI is that undesirable people could gain control of the content of the programs. Also CAI could cause the spread of existing poor courses and texts. The power of the computer to duplicate and distribute materials at rapid speed makes both dangers mentioned, readily apparent. According to Mr. Hicks, the teacher must be the watchdog to recognize these dangers. Another danger of CAI is that individual creativity could be denied if CAI is allowed to educate all students in a uniform method.

The moving force that inspired CAI was that of complete individualized instruction for each student. However, at this writing there are very few fully completed CAI courses. Until the time more fully developed CAI courses are available, complete individualized instruction will still be only a dream.

Mike Moser

CHAPTER VI

THE COMPUTER AS A CLASSROOM MANAGEMENT TOOL

BY

Jack Slingerland*

Janet Spitz

*Jack Slingerland, computer instruction specialist for Multnomah County I.E.D., served as group leader and editor for this chapter.

SECTION A: OVERVIEW OF COMPUTER MANAGED INSTRUCTION

With the advent of relatively inexpensive computer power, education has been faced with major decisions of how best to apply the computer to the problems of education. The early response was, and to some extent continues to be, to use the computer to speed up the data processing tasks traditionally associated with the operation of a school system. "For example, the initial applications of the computer have been designed to achieve more efficient scheduling of the same old course offerings, more efficient production of the same old report cards, faster scoring of the same old tests, and high speed access to computer-stored cumulative records with the same old teacher grades." (Cooley, 1971; pg 401)

While some progress has been made towards more imaginative administrative uses of the computer, particularly in the scheduling area, little has been done in the field of instruction.

The full benefits of the computer will not be realized until the computer has had a major impact on the instruction itself. In 1960 the computer made practical a massive survey of education in the United States. The Project TALENT survey (TALENT, 1965) dramatized deficiencies in present instructional methods for accomodating the individual differences of students.

How then might the use of a computer alter the instructional options? One approach has been computer assisted instruction (CAI) (See Chapter V). With CAI a student sits at a computer terminal and interacts with a computer which presents a lesson to him. While CAI holds some promise for the future, "...it is unlikely that, in the short run, it is going to make a significant impact on education because of the cost associated with one student utilizing a terminal for relatively long periods of time during each instructional session." (Dick, 1972; pg. 33)

An alternative to the use of the computer as an instructor is the utilization of the computer's speed and agility in processing large amounts of data to over-see the instructional process. Typically a computer managed instruction (CMI) system will involve, but not be limited to, the "mechanical tasks of scoring seat work and test papers, recording the scores, creating descriptions of pupils based on their scores, and keeping track of what instructional materials a student has used." (Baker, 1971; pg. 52) The major functional divisions of current CMI projects are the scoring of student responses, the diagnosing of educational difficulties, the prescription of instructional objectives and materials, and the reporting to teachers of student progress.

There are several large scale CMI projects currently in operation. Project PLAN, whose computer facilities are based in Iowa City, provides for record-keeping, test scoring, and instructional prescription in the subjects of science, mathematics, language arts, and social studies. Services in all four subject areas cost $100 per student per year. A more detailed discussion of PLAN is given below. In contrast to PLAN's daily interaction with a student's education, the Instructional Management System (IMS) developed by Systems Development Corporation is designed to provide weekly summaries and recommendations for remedial action on a student's measured weaknesses. Individually Prescribed Instruction (IPI), a product of the University of Pittsburg's Learning Research and Development Center and the Baldwin-Whitehall Public Schools of suburban Pittsburg, is a project which emphasizes the record-keeping aspect to support an individualized instructional program. The computer is used to collect and record information on each student and to summarize the information for teacher use. For a brief overview of these and several other CMI projects, the reader is directed to (Finch, 1971).

PLAN

Let us now direct our attention in some detail to three specific applications of CMI: 1) Project PLAN, 2) The Parkrose Computer Reading Program (PCRP), and 3) The Precise Behavior Management System (PBMS) as implemented by the nursing school at Mt. Hood Community College.

One of the first CMI programs was the Program for Learning in Accordance with Needs (Project PLAN) developed by John Flannagan at American Institutes for Research in the mid 1960's. PLAN, now marketed by the Westinghouse Learning Corporation, is one of the most highly developed CMI systems available today.

Project PLAN Is more than just a computer. It is a comprehensive educational system in which the computer plays a supportive role. The major components of PLAN are the teacher, instructional packages called teaching-learning units (TLU's), a list of learning objectives, objective tests,[1] achievement tests, and placement tests, all of which are inextireably related to the computer support system. With the exception of the teacher, all of these materials are provided by project PLAN. In addition, PLAN provides programs for training teachers and administrators who work with the system.

To date PLAN is available for grades 1 through 8 in the subject areas of language arts, mathematics, science and social studies. PLAN materials for the higher grades have been developed and tested but are not ready for release as of summer, 1973.[2]

---

1. "Objective test" as used in this discussion means a test which measures whether or not a student has achieved a particular instructional objective. It should not be confused with the use of "objective test" to describe a test which is not intended to be "subjective".

A look at the role of PLAN's computer is essential to understanding the system. A school using PLAN may either have a terminal consisting of a card reader and a console typewriter within the school building or it may share a "mini-center" with several other schools in the area. The information is transmitted via telephone lines to a central computer at the Measurement Research Center in Iowa City, Iowa. The information input may include student responses to placement and objective tests, homework, status cards filled out by the students indicating the objectives on which they are working, and the teacher's comments and observations. During the night, the computer processes the information given it that day and prepares a report listing the action to be taken. The report is ready for the teachers' use the next morning.

The daily printout includes 1) students' test results, 2) a report organized by learning objectives showing the status of those students who are studying a given objective, and 3) a report organized by student which includes the objectives each student has completed, those which he has started and the next three objectives scheduled to his individual program. The report organized by objective helps the teacher to schedule groups of students for discussion or tutoring, while the report organized

---

2. The school districts which participated in PLAN's development are:
   Bethel Park School District, Bethel Park, PA
   Hicksville Public School District, Hicksville, NY
   Penn Trafford School District, Harrison City, PA
   Pittsburg Public Schools, Pittsburg, PA
   Quincy Public Schools, Quincy, MA
   Wood County Schools, Parkersburg, W. VA
   Archdiocese of San Francisco, San Francisco, CA
   Fremont Unified School District, Fremont, CA
   San Carlos Elementary School District, San Carlos, CA
   San Jose Elementary School District, San Jose, CA
   Santa Clara Unified School District, Santa Clars, CA
   Union Elementary School District, Santa Clara, CA

by student allows the teacher to check each student's short-term progress and pin-point those who are having difficulty.

In order to produce the daily reports, the computer stores massive volumes of information. The various types of information in this data include 1) each student's academic and test records, 2) a list of each student's objectives and program of study, 3) a catalog of descriptions of all teacher-learning units, 4) a file of all test keys, 5) a record of the student's daily activities, and 6) relevant predictive and discriminative analysis data. The major role of the computer, then, is of an exceedingly competent secretary who frees the teachers to spend their time in contact with the students.

Each subject area is divided into segments of learning defined by objectives. Each objective is explicitly stated in accordance with the principles of behavioral objectives. When a student reads an objective, she should know what she is going to learn (objective) and how she will know when she has learned it (performance objective). A TLU can be used by a child without additional instruction or guidance from the teacher. For each learning objective there are several teaching-learning units available, each differing in the use of vocabulary, media, and learning style. The student's progress is measured on the basis of the same terminal test regardles of which TLU for that objective she has used. If a student does not test at the expected level, the computer will list the objectives not learned along with the questions missed. The student can then be recycled through the same material or she may be assigned to a more appropriate TLU.

Activities suggested in a TLU might include reading pages in a book, discussing a question with a partner, viewing a filmstrip, reading an Instructional Guide or working from Activity Sheets. (Activity Sheets and

Instructional Guides are part of the support materials supplied by PLAN.)

When the student has completed the teaching-learning unit, his progress can be measured in three ways: 1) Teacher Certification, in which case the teacher uses her professional judgement to determine whether or not a student has mastered the objective, 2) Teacher Evaluation using the PLAN criteria, or 3) an objective test taken by the student when he feels ready, and scored by the computer.

When a child first enters the PLAN program he is given an orientation to the system and the materials he will be using. He is then given a placement test. Based on his answers to the placement test questions, the computer provides a suggested list of TLU's to be included in his program of studies. This list may be modified by the teacher or by the student if either desires. This individual placement means that students have different courses of study, each tailored to the individual's achievement level and rate of learning. Placement tests are used throughout the year to assist in determining the student's course of study.

In addition to the objective tests which measure the student's mastery of individual instructional objectives, the teacher may use achievement tests to measure how well a student has retained those skills previously learned.

It is important to note that PLAN does not use any of its tests for the purpose of assigning grades. All tests are used solely for diagnostic purposes.

PLAN is a large and expensive system. It handles the full gamut of instructional management services for four core curriculum areas at a cost of [illegible] per student per year. Even at this price, which is well below the current costs of CAI, PLAN does not seem likely to be widely adopted in Oregon Schools.

PCRP

During the 1972-73 school year, a team of teachers from the Thompson Elementary School in the Parkrose school district working with the cooperation of the Multnomah County Intermediate Education District, developed a computer based approach to the individualization of reading instruction at the elementary school level. The design of the system is complete and the computer programs have been written. Field testing with students in the first six grades will begin in September, 1973.

When a new student enters the program, he is first given a written reading skills survey. His responses to the survey are scored by the computer to determine his reading level, the reading skills he posesses, and to isolate his reading deficiencies. Once the computer has determined the student's reading skills, it prescribes an instructional objective for him to meet.

With the prescription as guide, the student and the teacher decide how best to meet the prescribed objective. The teacher has a learning resources catalog keyed to each objective. The catalog lists available resources to be used to help the student meet the objective. These resources include:

1) A skill bank worksheet
2) A listening bank of cassette tapes
3) A game bank with a variety of games to support the concept
4) A filmstrip bank
5) A pencil and paper learning packet bank

When the student has completed those activities which he and his teacher have agreed upon, his progress is evaluated.

Evaluation is accomplished through the use of coordinated test materials and sense mark cards. The cards have been designed to match the format of the printed tests. The student merely follows an arrow from the test to the appropriate place on the card to mark his response to each question. [illegible] he completes the test the cards are read by an optical card reader in the school and the information is processed by one of MCIED's computers. Once the computer has evaluated the student's progress, a prescription for the student's next learning area is returned to the teacher and the cycle begins anew.

Summary statistics are also maintained by the computer for each student individually. From a terminal in the school the teacher may request a review of the progress of a particular student, or a review of the progress of all the students. This procedure allows the instructor to gain an overview of each student's progress and facilitates an individualized response to his particular needs.

PBMS

The Nursing School at Mt. Hood Community College is utilizing what they have termed the Precise Behavior Management System (PBMS). PBMS differs from PLAN and the PCRP in that it does not specifically prescribe instruction. PBMS's major contribution is that it gives detailed student profiles.

At MHCC PBMS is used for diagnosis in three distinct areas: the student, the instructor and the curriculum. The basis of this diagnosis is a large block of data on student performance. This data is collected by the instructors, the students, and the supervisors of student nurses. Once each 11 week term this information is summarized and transmitted by mail to the Behavior Research Company for analysis.

The primary focus of PBMS is the student. The data she amasses during her work is recorded on special graphic forms developed by Behavior Research Company. Between computer evaluations the data graphically displayed communicates a great deal about the student's progress and is reviewed bi-weekly by the student and her advisor. The quarterly computer analysis provides profiles of specific behaviors and also profiles of related blocks of behaviors which are then used for guiding the student's future course work to satisfy her individual educational needs.

The same data which is collected for generating student profiles may be used to provide a measure of the effectiveness of instructors. By examining the performance of students of a particular instructor, the strengths and weaknesses of that instructor may be identified. This information is used as feedback to the instructor to assist him in improving his teaching techniques and is also used in making teaching assignments that will maximize the effectiveness of a given instructor.

The curriculum itself may be examined using the student data collected. Correlations may be generated between desired student behaviors and specific instructional sequences to determine optimum teaching strategies. In a similar vein the data may be used to evaluate the effectiveness of various institutions in providing beneficial on-the-job training. From this evaluation the nursing school may decide which hospitals provide the optimum learning environments for the student nurses.

To obtain the extreme flexibility of PBMS, the students must devote a great deal of energy and time to data collection and Behavior Research Company must be paid for their analysis of the collected data. Currently the nursing school is spending approximately $35 per student per year for student behavior profiles.

PBMS provides detailed information about students, instructors, and the curriculum at a relatively low cost. As such it has a great potential for impact on the educational process.

CMI consists of a package which will test, diagnose, prescribe, and report to the teacher. We have examined in some detail a national CMI project and implementation of particular CMI systems in two schools in Oregon. What does CMI mean to the typical educator in Oregon? A candid answer would be "Probably very little at this time." But as the sophistication of the CMI systems increases and the cost of the implementation decreases, CMI will become a viable method for achieving individualization of instruction and more efficient and productive use of the professional educator.

Bibliography

(Baker, 1971)

Baker, Frank B. "Computer Based Instructional Management Systems: A First Look", Review of Educational Research, Vol. 41 No. 1 (1971), pp. 51-70

(Cooley, 1971)

Cooley, William W. "Computer Managed Instruction", Encyclopedia of Education: 1971, pp. 400-404

(Dick, 1972)

Dick, Walter and Paul Gallagher. "Systems Concepts and Computer-Managed Instruction: An Implementation and Validation Study", Educational Technology, (February 1972), pp. 33-39

(Finch, 1971)

Finch, John M. "An Overview of Computer-Managed Instruction", Educational Technology, (July 1972), pp. 46-47

(TALENT, 1965)

Project TALENT Staff. "A National Inventory of Attitudes and Abilities", (University of Pittsburgh) Project TALENT Bulletin No. 4, (February 1965).

Section B Abstracts of related articles for Chapter VI

A Student Centered Instructional System
Development Research School, Florida State University

The Student Centered Instructional System (SCIS) is a computer-managed system developed at the Development Research School primarily for use in the public schools. SCIS can be used for any course utilizing performance objectives, although to data emphasis has been placed on mathematics.

At this point five subsystems are utilized: 1) an initiation system; 2) a student record-keeping system; 3) a course analysis record-keeping system; 4) an instructional sequence system; and 5) a grading system. The initiation system is used to convert a conventional course to one based on performance objectives. The student record-keeping system provides the student and teacher with information on how the student is progressing, and also provides a method of accumulating data for a given student in relation to performance objectives. The course-analysis record-keeping system is a computerized monitoring system which provides teacher's, student's and parent's reports on the progress of the student for the current grading period. (A manual monitoring system may also be used.) The instructional sequence system is a sequence of activities used for each unit studies. The sequence consists of a pretest, assigned objectives, instructional material, teacher assistance (if needed), criterion test on objectives and a posttest over several objectives. The grading system is based primarily on the student's posttest scores and the number of objectives completed.

Assistance is available for schools wishing to initiate a SCIS, and instructional materials are available for seventh grade mathematics.

Janice Spitz

(NOTE: If you are interested in using any of the materials, contact Ms. Judy O'Steen or Dr. Janice Smith at the Development Research School, Florida State University, West Call Street, Tallahassee, Florida, 32306.)

Baker, Frank B.
Computer-Based Instructional Management Systems: A First Look
Review of Educational Research, Vol. 41, No. 1, pp. 51-71

Computer assisted instruction (CAI) was the forerunner of computer managed instruction (CMI). Several projects proved that CAI was feasible, but as yet it has not been shown practical. After the initial romance with CAI, research efforts have been directed towards CMI.

Much of this article "...describes a number of existing computer-based instructional management systems and ascertains their level of development." The projects described are the Instructional Management System (IMS) of Systems Development Corporation, the University of Pittsburgh Learning Research and Development Center's Individually Prescribed Instruction (IPI) and its related Management and Information System (IPI/MIS), the Program for Learning in Accordance with Needs (PLAN) developed by the American Institute for Research, the Teaching Information Processing System (TIPS) written by Allen Kelly of the University of Wisconsin, and the Computer Managed System (CMS) developed by the University of Wisconsin's Research and Development Center for Cognitive Learning to support the Individualized Mathematics Curriculum Project (IMCP). Some consideration is also given to the instructional management aspects of the CAI work done by the Stanford CAI project under Patrick Suppes.

Jack Slingerland

Griesen, James V.
Study of an Independent Study Program
College Management V. 9 (January 1974) pp. 30-32

New students at the Ohio State Medical School can choose a computer managed system of instruction as an alternative to the traditional lecture-discussion method. Both programs consist of three segments designed around body systems: Normal Man, Introduction to Pathophysiology & Therapeutics, and Pathophysiology of Man. The Independent Study Program (IPS), as it is called, combines computer study with one-to-one consultation with faculty (average of 3½ hours per week), lab demonstrations, group activities and seminars.

The program, utilizing an IBM 360/50 is organized into 34 modules, each written by an interdisciplinary team of faculty members. It is integrated horizontally, across the basic sciences and vertically, involving more clinical work with the basic sciences to make the beginning medical curriculum relevant. The individualized instruction makes this possible, affording the student more opportunities for clinical exposure as he works with faculty. Students take written and oral exams at various points in the curriculum and enter clinical clerkships when they finish the IPS.

A student paces himself through the program, speeding up or slowing down to fit his most effective learning rate. His progress is evaluated on an "average" 6-day week, 8 hours of study per day. If he falls behind "schedule" more than 50%, he must justify his case to a faculty committee. When he feels he has learned the material for a module, he goes to a terminal to interact with the program's Tutorial Evaluation System (TES) which asks questions and evaluates the answers, indicating to the student where his deficiencies lie, prescribing additional study assignments, reviewing previous material or suggesting faculty conferences. Students keep the printouts for reference. Faculty may obtain up to the minute reports on individual students and groups who have finished particular modules, which simplifies scheduling of other activities. The computer also provides an item summary of responses to the TES as a basis for improving item construction. Student suggestions for improvement are also stored and made available to the faculty.

IPS is not new at Ohio State Medical School, the first program being initiated in 1962. The initial program proved self-defeating as it existed in conjunction with the traditional program; it also lacked adequate evaluation processes. The school secured a three-year grant totaling $1,088,000 from the Bureau of Health Manpower Education of the U.S. Public Health Service, National Institutes of Health, to design a total program. Ohio State has what is probably the only medical curriculum in the country that can be picked up and used by other schools--a boon for new medical schools. Some twenty other medical, nursing and dental schools have requested Ohio State's program, indicating that the interest in IPS is strong over the nation.

School officials predict that IPS will be an aid for minority students unaccustomed to highly structured instruction or those who have weaknesses upon entering medical school. The program is highly expendable, allowing addition of students without rewriting programs and purchasing new equipment except for terminals. As the program progresses, faculty are becoming more adept in writing programs, an initial problem. The author admits the initial high cost of TES but says the costs level out over a period of time. Ohio State Medical School operates its IPS program on the theory that personal motivation and self-discipline produces students with a desire for life long learning, an important requisite for modern practitioners of medicine.

D. Vest

Gibb, Glenadine E.
The Computer--A Facilitator in Management and Instruction
Mathematics Teacher, Vol. 9, No. 11 (January 1973)
Pages 17-21

There seem to be two general directions computers in education are to take: (1) instructional individualization through computer-managed instruction (CMI) and computer-assisted instruction (CAI), and (2) the use of t-e computer as a computational device and as a means of simulating concepts within the present curriculum.

In most attempts at individualized instruction the teacher can easily wear himself down with clerical work, and will often go back to the traditional "page by page" presentation of material which allows for little concern for the student's ability to master the material. Much of this clerical work can be handled by a computer, which can test, score, diagnose, and prescribe remedies for deficiencies. This relieves the teacher of the mammoth clerical burden associated with teaching and frees him or her to spend more time in the planning of individualized instruction.

The computer may also be used to assist in the presentation of simulated problems and knowledge beyond the scope of the core program. CAI provides a tutoring system in which exchange of questions and answers takes place between student and machine. The immediate feedback and undivided attention that the machine can give the student may be the critical need of an individual student at any given time. Caution must be exercised in remembering just how little imagination the computer itself has. It must be programmed by someone with a great deal of knowledge and experience, and therefore much time and effort must be provided by someone to "instruct" the computer so it, in turn, can instruct the students.

With respect to its use either as a facilitaror or manager of instruction, the computer is a means of supplementing, not supplanting, the teacher, making it possible for him to take on the human part of teaching. Our professional education curricula must be drastically changed if teachers are to be prepared to assume this new role where they must shift much of their focus from drill and clerical work to responsibility for instructional design and strategy.

James F. Muck

Hansen, Burdette
The Computer in Education
The Clearing House, Vol. 45, No. 4, (December 1970), pp. 195-200

Education is under attack from many sources. Taxpayers are revolting against higher and higher costs and also against the fact that there is no effective way to judge the finished product. Education is being called upon to be more responsive to the individual - that is, to take the ever-expanding body of knowledge and fit it to the individual, rather than trying to fit the individual to the knowledge.

A remarkable tool called a computer can help education meet these demands. The computer is being used now in several areas. One of these is in scheduling. Many high schools have over 100 course offerings, and it used to take hundreds of man hours to prepare a master schedule with this many course offerings. This job can now be turned over to a computer.

Another way in which the computer is helpful is in individualizing instruction. Very complex modular scheduling is now possible. The computer helps determine the best of millions of possible combinations of classes; and by scanning the backgrounds of the staff members, it can help to determine the best people to teach these subjects.

The computer is very useful in evaluation educational output. It can help answer questions such as how we as a nation are attaining agreed-upon educational objectives, what progress we are making in American education, and what problems we still face.

The computer is useful in evaluation because it makes possible the inclusion of large numbers of factors in the analysis of a system. But perhaps the most basic way in which the computer helps in evaluation is that it forces people in education to think systematically and with clarity about their methods and goals.

In a cost-benefit analysis, a computer can be very helpful. The reason, again, is its capacity for allowing the input of huge amounts of data. This makes it possible to arrive at better decisions about which items to buy in a limited budget situation.

Great strides have been made in the areas of computer assisted instruction (CAI) and computer managed instruction (CMI). With more sophisticated CAI programs, instruction can truly become individualized; because of high costs, though, we are a good ten years away from widespread use of CAI. Computer managed instruction, however, is becoming quite practical. An example of this, project PLAN - Program for Learning in

Accordance with Needs - is operating in sixty-three school districts across the country.

Some additional uses of computers could be information storage and retrieval, to free teachers from routine clerical work, and keeping up with technology. It took hundreds of years for education to feel the effects of the printing press, but it will not take so long to feel the effects of the computer.

John Shirey

Hoffman, Jonathan
How a Computer Helps a Teacher to Teach
School Management, (October 1972), pp. 16-18

The Providence, Rhode Island, school system is utilizing an automated record keeping system to gain precise information about students' progress. The system, implemented on an IBM 1130, was jointly developed by the Providence School District and National Educational Program Associates, a Boston consulting firm.

The initial project was developed for a Model Cities reading program for students in grades one through eight. The program was designed around a list of 124 reading skill objectives. The program currently involves students in six schools. The first year of operation for these schools showed all grades with gains from four to twelve months greater than predicted.

The computer is used to store information about each student's progress, the time taken to achieve an objective, and the materials used in that learning process. From this information is generated objective attainment analysis lists, progress reports for distribution to parents, a materials-effectiveness evaluation, a monthly materials cost analysis report by objective, a yearly summary of materials cost and teacher time spent on each objective, and summaries of standardized test results for each teacher's class.

The introduction of the new reading program called for extensive inservice training for teachers. It also has changed the role of the administrator: "...administrators, while not instructors, have become legitimate instructional leaders. As the monitoring system signals teachers and parents, so it involves the school administrators more closely than ever in the teaching-learning process."

Jack Slingerland

Lawler, R. Michael
Selected Instructional Strategies in Computer-managed Instruction
Florida State University, Tallahasse, Computer Assisted Instructional Center, 1972, 30 pages

This article reports on the results of a study of the differential effects of three instructional strategies in a computer-managed instruction (CMI) environment. Subjects were 167 undergraduates in a health education course at Florida State University. Criteria for instructional success included performance, attitudes, and time spent.

Forty-one students received traditional classroom instruction (CI) and served as a control group. The remaining students were randomly assigned to one of three CMI treatments. For these latter students, the course was divided into 14 modules with a total of 32 objectives. Throughout the course, module posttests (5 items per objective) were administered; the criterion for passing a module test was set at 80 per cent. During the first week of classes the pretest and attitude measures were administered to all students. For the remainder of the quarter, the CMI students proceeded with their study of appropriate self-instructional materials. When a student felt prepared, he scheduled time on a CAI terminal and was administered the posttest on the module he had completed. Upon course completion, the attitude measures and final examination were administered to all students.

Of the CMI treatment groups, the Remedial Prescription-Forced Mastery represented the most typical CMI strategy. If students in this group failed to reach criterion on an objective, they were presented remedial prescriptions and were required to take another randomly chosen set of items until they reached criterion. Students in the Remedial Prescription Forced Progression group who failed to reach criterion on an objective were presented only the remedial prescriptions and were not permitted to repeat the failed posttest. In the Forced Progression group, students who failed to reach criterion on an objective were given neither remedial prescriptions nor were they permitted to repeat the failed posttest.

The results demonstrated a general superiority of the CMI groups on final examination performance ($p. < .05$). This superiority was attributed to (a) a greater degree of familiarity with the objectives and criteria, and (b) possible differences in levels of achievement motivation on the final examination. Among the CMI groups, the "forced mastery" groups did significantly better on the final exam ($p < .05$). No significant differences were demonstrated between the four groups in terms of attitude change. Nor were there significant differences in study time or in the number of days required to complete half or all of the module posttests. Average time on the computer for all CMI students was approximately 3.5 hours, while CI students spent 30 hours in the classroom.

Oscar V. Evenson

Walters, Richard F.: Nishimoto, Gail M.: Horowitz, John M: and Burger, Ray E.
Initial Development of Individualized Instruction with Computer Support
Proceedings of the 1972 Conference on Computers in Undergraduate Curriculum, published by Southern Regional Education Board, Atlanta, Georgia 30313

The gap between large instructional projects such as project PLAN and what a small group of teachers might hope to design and implement is appallingly great. The authors describe what one group of teachers has done to implement a small instructional management that provide self-evaluation for the student, course review for the instructor, and an opportunity to proceed into more sophisticated computer-supported instruction."

The system consists of three programs. One is for entering the answer key and related comments. A second program is the one which the student uses when entering his answers. The last program keeps statistics on students' responses and prepares reports for the instructor's use.

Under this system the student enters his answers to a printed question list. The responses are scored and, at the instructor's option, correct answers are presented. Individual comments from the instructor to the student are also printed.

The article contains sample outputs from each program and offers suggestions for future developments of these programs.

Jack Slingerland

Willis, Mart[illegible]
DOVACK's Machines Help Children Read,
American Education, (June 1971),
Pages 3-8

DOVACK stands for differentiated, oral, visual, aural, computerized, Kinesthetic. It is in the experimental stage in a community in northwest Florida. DOVACK is being used with students who are way behind in reading and vocabulary. It works as follows. The student dictates a story to a tape recorder. Overnight, that story is punched onto cards and fed into a computer. It is typed exactly as the child dictated it, bad garmmar and all. The next day, the child may sit and listen to the story he dictated the day before. He is also provided with a printout of his story which he can read as he listens to it.

The DOVACK student spends one class period a day in the program. The first thing he does during this period is to take out his folder which contains previously dictated stories as printed out by theCControl Data 6400 computer located at Florida State University. "Each time he dictates a story, a DOVACK teacher or aid listens to it as she punches it onto cards. She then redictates it for him, replacing his strong Afro-American dialect with standard English pronounciation without changing the original word order."

Before the student does anything with these recordings and printouts, he dictates a new story. Preferably, he dictates something he has recently experienced--a television show for instance. The use of his own experience is DOVACK's core philosophy.

Now, the DOVACK student continues with his agenda, which can consist of a wide range of activities. He reads the printout of the previous day's dictation, listening with his ear phones to his teacher's voice carefully pronouncing the words. The DOVACK student might then choose an older story from his file and read it aloud to another student or a group of students. Or he might listen to another student read to him.

Bad grammar is not ignored forever. From several of the child's stories, a composite story in the third person is made, using correct grammar. The child reads this story aloud and studies it as another way of putting things he had said earlier.

Tests are administered every six days on the new words the pupil has introduced since the last exam. Every thirty six days the student is tested for recall. Progress reports are printed along with a scatter chart of the scores of all students to assist the teacher in viewing each students progress with respect to the entire DOVACK population.

It appears on the surface that DOVACK is one successful way of dealing with reading problems. However, it is not the only way. Much raw data needs to be analyzed before any unqualified claims can be made. However, the experiment does show that DOVACK should be made economically feasible to anyone who wants to use it.

John Shirey

CHAPTER VII

ADMINISTRATIVE USES OF COMPUTERS

by

Ed Anderson*

Paul Ashdown

Ken Bierly

Mike Dunlap

David Hoerger

John Lewis

Michael Moser

Larry Nesland

*Ed Anderson, College of Education, Oregon State University, served as group leader and overall editor for this chapter.

SECTION A: Administrative Uses of Computers

Overview

It is becoming increasingly clear that we stand at a crossroads in education. This is true in all areas that are usually considered integral parts of this concept - teaching, learning, the learner, curriculum, evaluation and measurement, philosophy, and administration - to name a few. We are in the midst of one of the greatest revolutions in recorded history, the technological revolution. The explosion is chronicled all around us, whether we consider organ transplants, space exploration, or communication systems. A prime mover of this revolution is the computer. A baby's cry in California is diagnosed by computer. A bank in Washington offers computer printouts showing customers their cumulative budgetary expenditures by category. A college in New York experiments with shopping by televised computer hookups. Heart diseases are detected, complex space voyages are controlled, school enrollments are projected, all by computers. It is our belief that within our lifetime, the computer will significantly alter the life of every human being on this planet. The power exists; it will grow. Someone will control it. Someone will make decisions (computer based) that will affect you, your schools, your students, and your teachers. If educators are to have a voice in the uses of computers in the educational process, we must learn about the computer and consider the implications of the computer for education.

Educational Administration. The contributors to this chapter on administrative uses of the computer faced a basic problem. What is administration? What tasks performed by computers can be called administrative?

There are several well known models or definitions of educational administration. Getzels and Giba have characterized administration as a role in a social system. Knezevich and others discuss administration in terms of a series of tasks, including such things as curriculum, business, pupil personnel services, facilities, and public relations. Fayol, Gulick, Litchfield, Sears and others have proposed a model of administration as a process involving planning, organizing, coordinating and evaluating. Some authors view administration as various combinations of these and other definitions.

It is clear, then, that different readers of this section will view administration differently. Add to that problem the fact that elementary and secondary school administration have important differences. How can we consider everyone's view? In attempting to solve this problem, acknowledgement is made of these different views. Therefore, the content of this section has been written for everyone making administrative decisions and engaged or interested in administrative (organizational leadership and management) activities.

Problem Areas. There are certainly economic deterrents which prevent the use of computers in school administration. Limited school budgets are already stretched just to keep up with the status quo. It is difficult to justify large expenditures of money for computers unless significant positive results can be demonstrated. Moreover, money used for administrative purposes is not available to use for instructional purposes. We must admit that it is difficult to not justify these large expenditures for tools, teachers, curriculum, methods, structures which improve the instructional process. To justify administrative use of computers involves careful planning, design, implementation and evaluation (many of these topics are discussed in other areas of the handbook). It is difficult, if not impossible, to demonstrate for and convince school boards and administrations of the practical and future uses of the computer unless they are willing to be educated in this area. All educators, not just administrators, need to explore any method or concept that will enhance our educational system.

In addition to economic considerations and the need to know more about the computer, there is a serious problem in the preparation of personnel qualified specifically to design, establish and operate educational data processing systems in school districts. If computers are to be seriously considered for educational uses, we must prepare people in this line of work who have had experiences in the classroom and in school management. No longer can we allow people and agencies outside of educational circles to control education in terms of computer applications, textbooks, standardized tests, etc... It is time for educators to direct these activities. This does not imply that we cannot learn from the experience of others. We certainly should and must do so. This does imply that educators are responsible for education.

Why Computers in Administration? To balance, or overbalance, the deterring factors already mentioned, there are factors promoting computer technology in schools. Administrators are certainly looking for ways to release teachers from clerical and routine paper work. Also, they are seeking and using types of information never collected before which is now deemed necessary to make decisions concerning the efficient and effective operation of a school or district. For improvement of curricula, rapid feedback, for the diagnosis of student needs, assessment of results, and for efficient instructional use of diversified materials, automated cataloging and retrieval systems are necessary. Innovations call for computers; computers, in turn, facilitate innovations.

What are some conclusions to be drawn concerning the use of the computer in administrative tasks? Ample evidence abounds to support the application and acceptance of the computer when it is used for masses of data related to well-defined tasks, when processing is highly repetitive, when rules for decisions are specific, when processes are to be repeated many times under a variety of conditions, and when there is great demand for speed. When computerization has been appropriately applied to educational data there is little doubt that it has reduced the amount of professional time and energy previously devoted to clerical work; it has reduced the unit cost of data processing and it has facilitated the development of new techniques, resources and processes not before feasible with previous technology.

One of the most valuable (some would question this) results from introducing computer technology into education has been the need to examine more precisely the nature of learning itself and the accompanying procedures, schedules, objectives and assumptions. Put a different way, the computer is a valuable tool in the study of the nature of learning and the educational process.

Technology changes the role of the administrator. As the administrator places increased reliance on the computer and automated procedures, he or she must understand what this implies. A computer can supply the school administrator with more comprehensive and up-to-date information. However, access to more data does not ensure wise selection or use of it. The most striking feature of computer technology is the wide gap between practice and potential. This disparity applies both in the nature and scope of applications and in the breadth of understanding and acceptance of the power of these applications to improve education. Computers cannot eliminate human failure, but they can enhance success of human endeavor.

Remainder of This Chapter. The following reports include discussions on and/or examples of administrative uses of the computer. The first section covers the current state of the art. This area has been delineated into five divisions of administrative tasks; finance, facilities, personnel, curriculum, and students. Many of the tasks listed are currently being performed for administrators whose districts have contracts with OTIS (Oregon Total Information Service).

Accountability is a keyword to today's administrator. The section on Management Systems discusses three models; PPBS (Planning Programmed Budgeting Systems), PERT (Program Evaluation and Review Techniques), and RRPM (Resources Requirements Prediction Model). These technique analyses offer the administrator a tool with which to plan, implement, and evaluate programs and resources.

The next section briefly covers the counseling aspects as an area which receives considerable administrative attention. This is followed by some comments intended to give a brief future perspective. A list of resources and a selected bibliography concludes this chapter.

The programs and examples which are listed and/or discussed do not encompass all current uses of the computer. The intention is to inform the reader of some current activities involving the computer in the administrative field.

General Administrative Uses

Computer technology is advanced far beyond actual school administrative uses. Moreover, educators are far behind in their ability to define their needs and problems, and in turn, have not communicated to the computer system designer what they want. Present hardware can be programmed to do a great many of the listed functions of administration. It should be kept in mind that the computer is a tool that relieves people of tedious projects and assignments. It can perform these tasks faster and more accurately with greater economy than is possible with people.

The administrative functions can be listed under five headings: 1) Finance; 2) Facilities; 3) Personnel; 4) Curriculum; and 5) Students. In all of the functions listed below the computer can play a significant role, either assisting the school administrator in the task, or handling the task completely.

Finance.

A. Budgeting and Accounting

1. Making analysis of estimates and requests for funds.

2. Allocating funds to specified categories and keeping up-to-date bookkeeping system with a constant analysis of projected spending compared to actual spending.
3. Projected cost and estimated incomes with analysis of the total overall budget.
4. Printout of financial reports to various levels of responsibilities i.e., board, state, and federal agencies.

B. Payroll

1. Preparation of payroll with automatic printing of checks.
2. Maintenance of accounts of each employee for quick retrieval and analysis.
3. Preparation of reports to tax, retirement, and insurance officials as well as the employee.
4. Salary accounting from various budget categories and to various educational programs.
5. Complete analysis of payroll expenditures with regard to budgeted amounts.
6. Preparation of proposed salary schedule and analysis of input on overall budget.

C. Purchasing

1. Accumulation of requisition for quantity purchasing.
2. Automatic control of encumbrance and release funds.
3. Maintains complete analysis of materials and supplies from vendors along with all statistical information.
4. Make available an analysis of disposition of goods and services purchased.
5. Preparation of requisition procedures for stock items with automatic recording.
6. Makes allocations of costs to budget categories and projects.
7. Prepares routing and delivery schedules for instructional and office supplies.
8. Continuous up-to-date inventory of supplies and materials both in use and stocked.
9. Generation of bill and general debiting and crediting to proper accounts with immediate follow up of unpaid bills.
10. Preparation of food ordering, inventory and payment of accounts.
11. Analysis of available food items and price in relation to diet and pupil eating habits.
12. Projected cost and income of food services.

All of these functions can be and are being provided by computer utilization. Not every school district involves the computer in these areas of finance; however, in large districts the economics provided by using up-to-date data processing are worth investigating. Keep in mind, though, that any and all of the areas of finance can be and are being handled by people.

Facilities.

A. Maintenance

1. Scheduling of preventative maintenance.

2. Replacement scheduling and cost appraisal of building items.
3. Cost analysis of product reliability and durability.
4. Repair scheduling with cost compared to replacement of items for better decision making.
5. Depreciation schedules.

B. Future Building Needs

1. Projected building need based on student population and community expansion.
2. Up-to-date analysis of building costs and projected costs.
3. Preparation of budget and election materials for community support.

Personnel.

1. Job evaluation and analysis of need and performance.
2. Complete records of staff with up-to-date experience and qualifications.
3. Salary information and replacement.
4. Personnel selection information from interviews with up-to-date comparisons of candidates qualifications.
5. Personnel evaluations.
6. Personal files of staff status with respect to district, state, and federal requirements.
7. Scheduling of teacher loads and classroom assignments.

Curriculum--Registration and Scheduling. Scheduling and registration have been two of the first areas of application utilized by most public schools and has considerable acceptance among administrators.

Reasons for the use of computers in registration and scheduling include: 1) business data processing equipment is well developed and suitable; 2) large increases in enrollment--with wide varieties of student needs, highly varied curriculum, and course offerings often exceeding 100--have caused tremendous clerical burdens; and 3) it allows a school to carry out routine information processing with increased speed and accuracy.

Basic obstacles in computer application in registration and scheduling have been: 1) cost, 2) lack of adequately trained educators to plan and supervise the activities, and 3) lack of good access to adequate computer facilities.

In scheduling and registration, the administrator must gather information regarding his staff, the facilities to be used, the course offerings, student course selections and other information on how the school wishes to conduct its business. The information is fed into a computer which collates the material and prepares reports of use to the school administration. These may include:

1. Records of course requests.
2. Analysis and comparison between requests and actual proposed program.
3. Check for course prerequisites
4. Listing of students requesting special courses.
5. Conflict schedules
6. Master schedule preparation.
7. Simulation of the master schedule for purposes of refinement.
8. Class schedules.

9. Counselor assignment lists.
10. Temporary and varified class lists for teachers.
11. Room utilization and room assignment lists.
12. Extracurricular activity time assignments.
13. Locker assignment and lock combination records.

With this information available, the administrator can more easily analyze teacher loads, room loads, and student demand for courses. He can better answer questions such as, "How many courses are available in each field and at what levels?" and "How many and what kinds of students are taking these courses?"

An area just beginning to be explored by administrators is that of projecting enrollment changes. Questions such as, "Will the school district's enrollment continue to grow (or decline) at the same rate it has in the past several years?", "Do I know how many children, by grade level, I can expect from those new housing developments?", and "Should I build more permanent classrooms now? What about five years from now?" are typical. A good school management information system can provide answers to such questions. It can aid in the exploration of answers to these and similar questions under a variety of assumptions of possible changing population.

Students--Census, Attendance Accounting, Progress Reporting. Student census is composed of facts regarding age, sex, grades and grade level, interests, test scores, attendance, health information, and socio-economic background of a pupil.

Through analysis and/or correlation of different items and factors stored in data files, and administrator can obtain:

1. Enrollment projections and pre-school surveys.
2. Federal employment of parent surveys.
3. Attendance law compliance information.
4. Student directories; addresses, phone numbers, etc.
5. School directories, district directories, and family unit directories.
6. Mailing labels for mailing school communications.
7. Bus transportation and schedule control lists.

Attendance accounting provides the capability to collect and maintain information needed for the state attendance report.

Available feedback or output for the administrator includes:

1. Daily attendance bulletins.
2. Period by period attendance accounting.
3. Weekly attendance report.
4. State quarterly report.
5. Annual report.
6. Recording, posting, and summarizing; by pupil, by classroom, by school, and by district.
7. Listing of students with possible attendance problems.
8. Listing of students on suspension, with expected dates of return.
9. Listing of withdrawn students.

Pupil progress reporting provides for collection of student performance marks and the updating of the student's record in the data file.

Available output for an administrator includes:

1. Printing of report cards.
2. Exception listing (those not given one or more grades).
3. Honor Roll--GPA
4. Accumulative Honor Roll--GPA rank within class.
5. Grade listing--year to date of courses and grades received.
6. Pupil transcript.
7. Grade analysis--by individual course and by individual teacher.
8. List of deviations from expected performance by student.
9. Progress reports--high and low achievers.
10. Correlations with ability groups.
11. Underachiever identification.
12. Computation of various averaging and ranking statistics.
13. Standardized test scoring--student profiles, score lists, frequency distributions, and correlations.

The ultimate purpose ond objective of all decisions in education should be related to the learner. Therefore, student data is the pivot around which all the rest of the data and all administrative decisions must be based.

Since a wide variety of background information concerning each student can be collected and accessed, one must avoid possible invasion of an individual's rights. One must make sure that data is not indiscriminately put into pupil files and that access to these files is carefully regulated.

Administrative uses of the computer are certain to expand within the next few years. A major trend is the development of total integration systems, capable of gathering information from any of the five areas previously mentioned, and, on request, presenting it in a meaningful report to the school administrator as background for decision making. Specific examples of such systems are OTIS (Oregon Total Information System) based in Eugene, Oregon and TIES (Total Information Education System) in operation in Minnesota.

Management Systems

As schools become larger and more diversified, it becomes necessary to develop tools for management of the system. Computers can and are being used by schools to facilitate planning, developing programs, and budgeting of programs. In this section a planning-programming-budgeting system is defined and explained and the roles that the computer is playing and will be playing, in the total management of schools is discussed.

PPBS. A Planning-Programming-Budgeting System (PPBS) is a major system analysis technique enabling school administrators to identify objectives, delineate programs to achieve objectives, analyze alternatives, allocate resources over a specific period of time, and compare costs and effectiveness.

Program Planning Budget Systems refer to output oriented programs and activities of an organization to specific resources that are normally stated in budget dollars. Both programs and resources are projected for at least five years in the future. Emphasis is placed upon outputs, cost effectiveness and methods, rational planning techniques, long-range objectives, and analytical tools for decision making.

The concept of PPBS was introduced to industry and government in the early 1920's. However, the technique of relating budget costs to program Output was not recognized as a powerful management tool until the Hoover Commission in 1948 recommended its implementation in government agencies.

PPBS became an operating reality in 1961, when Robert McNamera required its implementation for budget management in the Department of Defense. The successful implementation of PPBS in the Department of Defence inspired President Johnson in 1965 to issue an executive order to the effect that all federal government agencies would be required to change their function-object oriented budgets to a program accountability focus. With the pressure from society on educational institutions to be accountable came the presence of PPBS in public education. During the past five years, educational institutions have carefully and slowly approached the utilization of PPBS, and in some districts experimental programs have been established. The implementation of PPBS techniques have been hastened during the past two or three years because state legislatures are beginning to carefully scrutinize the output factors in education in relation to increased appropriations. The outgrowth of this increasing legislative concern is the need on the part of educators to demonstrate accountability in terms of substantiating requests for funds through the development of standards or measures by which educational output can be measured.

As of 1972, approximately 1,000 school districts in 30 states had achieved uneven rates of success with PPBS, e.g., Dade County, Florida; Skokie, Illinois; Pearl River, New York; Darien, Connecticut; Philadelphia, Pennsylvania; and Shoreline, Washington. All of these schools have some form of PPBS, although it may be given a different name. Locally, in Oregon, there are several schools beginning a PPBS program, e.g., McMinnville and Cottage Grove. The program in McMinnville, Oregon is referred to as PPBES (Program, Planning, Budgeting, Evaluation System). In Cottage Grove the PPBS program is referred to as SPECS (School Planning, Evaluation, and Communications System). Although there are many different names for PPB systems they are all basically similar. The future of PPBS seems promising in that it offers schools an opportunity to design a long-term plan for creative instruction. In addition, PPBS helps relate the cost of a program to its merits and links all areas of schools such as teacher aids, supporting activities, research, libraries, administration, research and development of subject matter in terms of time and cost.

Specifically, most PPB systems assist in answering the following questions:

What objectives should the school achieve?
How effectively are existing programs achieving them?
What are the long-term plans?
What resources are needed and how can they be allocated to best advantage?

Are there promising alternative approaches to securing better results?
Are the funds the public supplies for schools a good investment?
How could the return be improved?
What priorities should be established among the goals and programs?
Should some activities be eliminated? Which ones?
What activities should be given greater support?
Should taxes be increased?
How will additional revenue be spent?
What specific benefits will result?
Should some of the current responsibility of the schools be turned over to other agencies?
Should the schools assume additional functions to fulfill unmet needs? What are they?

The first step in a PPB system is listing what the school system expects to do. The objectives must be stated in clear, precise, and as far as possible, measurable terms so that they can be used as bases for evaluating the effectiveness of educational activities. The school then proceeds to explore alternatives for achieving its objectives, chooses the most promising and feasible approaches, analyzes its program needs, and allocates resources necessary to carry out learning activities needed to achieve the goals. Evaluation, inherent in PPBS, reveals the extent to which the objectives supported by the allocated resources have been achieved.

Through the consideration of alternatives, the process encourages innovation, personnel involvement, and program improvement. The total educational endeavor must be integrated. The line-item concept of budgeting must be replaced by a new way of thinking and of allocating funds. If for example $15,000 is budgeted for technical assistants, the budget must show precisely the purpose the expenditure will accomplish. Input is determined as it is needed to produce desired output.

The computer is an essential ingredient in the program planning budget process. Its capability to carry out alternative sets of computations rapidly provides the core for decision making. Since the PPBS process requires many people in the organization to be conversant with objectives, programs, and program costs, the output generated by the computer provides a clean communications link between every level. Also, it is well to note that program planning budget systems require a considerably greater amount of data than do traditional budgets. The computer with its data storage capabilities and rapid retrieval abilities serves a very useful purpose in the management of this dimension of PPBS. The very future of PPBS depends on the ability of organizations and schools to make use of computers and their capabilities.

PERT. Program Evaluation and Review Techniques has received widespread attention in government and industry. It appears that there may be application of PERT to educational problems and activities of a varied nature, including school construction, research projects of all types, and budget planning. Essentially, PERT is a management control technique which may or may not be computerized.

The basic procedure is to establish an ordered sequence of events to be achieved (network or flow plan). Associated with each event is one of three time estimates; optimistic, most likely, and pessimistic. The technique allows for probabilities to be attached to reaching intermediate events and the end objective on schedule. The probability is a calculation of what people think about the future and not a calculation of the future itself.

The technique is further refined by additional input concerning progress toward an identified objective on a regular time basis. The three basic outputs of a computer PERT analysis are:

1. The expected time of completion of each event.
2. The identification of slack and critical areas in the programs.
3. An expression of the probability of equaling or meeting the current schedule and the specification of the latest date by which every event must be completed in order to meet the end objective.

According to Kaimann (1967):

> Educational administrators are finding increasing evidence of these control techniques applied to construction projects. Alert administrators are extending the educational application to include those large nebulous research projects supported by grant funds. A number of local administrators are applying PERT to small but essential planning functions. All told, evidence is rapidly pointing toward a wealth of applications of PERT for better, more efficient, and more precise administration at all levels of educational administration. (p. 243).

RRPM. The Resource Requirements Prediction Model is an instructional cost simulation model for use in all types of post-secondary institutions including community colleges, vocational schools, and large and small four-year institutions with or without major research activities.

Development of the RRPM at the National Center for Higher Education Management Systems (NCHEMS) has been a long and arduous process. Many experimental prototypes were developed along the way; many concepts were tried and evaluated. The latest RRPM version (RRPM 1.6) has resulted from the work and contributions of many people over the past several years. The model is supported by the Western Interstate Commission for Higher Education (WICHE).

The concept of cost simulation in higher education has received considerable attention over the past several years. A primary purpose of RRPM is to generate information necessary for the preparation of instructional program budgets. Institutional data, either historical or projected, may be put in the model. RRPM calculates program cost information and resource requirements needed to undertake a given series of programs.

Another purpose of RRPM is to provide institutions with a flexible tool to analyze various institutional alternatives for utilization of a limited set of resources. The model has been designed as a long-range planning tool to aid higher-level management in rapidly determining resource implications of alternative policy and planning changes. Employing the model in this experimental mode, the user may ask a series of "what if" questions related to admission policies, curriculum changes, etc.

For those institutions with the analytical and programming capability RRPM provides a point of departure for their own modeling efforts. It is hoped that RRPM is sufficiently flexible to permit adaptation to specific institutional requirements without modification of the computer programs. However, some institutions will want to change the format of reports or other items. Toward this end, the model has been constructed with a modular scheme that makes modification of reports, etc., relatively convenient.

RRPM 1.6 generates four different types of reports, any or all of which may be requested by the user. These types are: 1) organizational unit reports providing line-item budgets for various organizational units within the institution, 2) program budget reports indicating the discipline or department contributions to various instructional programs, 3) institutional summary reports, and 4) formatted display reports that show all parameter data for the institution.

Computers in Counseling

Counselors in our schools need comprehensive, accurate, and up-to-date data if they are to be effective in assisting students to make decisions. Many times the data is needed immediately to capitalize upon a critical moment in the counseling session. New means of collecting, processing, and accessing information have been developed in the past few years. It is becoming evident that the computer can contribute significantly to the general area of guidance and counseling. Computers can store information concerning vocations, careers, and college admission requirements in addition to storing, organizing, calculating, analyzing, and outputting data into a meaningful form for both the counselor and student.

The role of the counselor varies from district to district and school to school. In many school situations, the counselor is a giver of information and is not a trained person in handling emotional and psychological problems. If all counselors were trained in the latter area and fulfilled that role in the school system, the computer could certainly be integrated into the system as the preliminary information giver in the area of careers, requirements, test results, etc. However, it may not be too long before the computer and the student may interact concerning personal problems which the student may not wish to discuss with the counselor. This type of program is for the most part still experimental, an example being ELIZA, developed at M.I.T. in 1966. This program permits certain types of dialoque between human and computer. The interactive dialoque between human and computer is a key area of research in the field of artificial intelligence, and is also important in computer assisted instruction.

Present Systems for Vocational Guidance. The U.S. Office of Education reports (1969) several systems in the planning or implementation stage. Those reported at that time were:

The Harvard-Needs-Newton Information System for Vocational Decisions
The IBM Experimental Educational and Career Exploration Center
The American Institutes for Research Comprehensive Vocational Guidance System
The SDC Vocational Counseling System
The Rochester Career Guidance Project
The Pennsylvania State University Computer-Assisted Career Exploration System
The University of Oregon GUIDPAK system
The Willowbrook Computerized Vocational Information System
The Palo Alto Computer-Based Course Selection and Counseling System
The Bartlesville Total Guidance Information Support System

All of the above systems include programs involving information retrieval. Computers excel at this task and complete the job faster and more objectively than any other method. A complete (ideal) system should provide four main support benefits for the counselor:

1. Instant information to support the counseling interview.
2. A rapid diagnostic capability which eliminates human time in searching, calculating, interpreting and analyzing data.
3. Instructional type capability to provide information to students in a vast number of areas.
4. A synthetic confrontation therapy dialogue for those students who prefer to have a non-human entity listen to these personal problems.

Since its beginning the University of Oregon GUIDPAK system is widely used in Oregon, and is representative of what can be done in this field. Since its beginning, its name has been changed to C.I.S. (Career Information System), and will be referred to henceforth in this section as C.I.S. The project was funded by the Manpower Administration of the Department of Labor in 1969 to help solve the problems that vocational guidance counselors are faced with today. C.I.S. is a statewide agency which provides current labor market information to schools, social agencies, and individuals in Oregon. The system is both computerized and non-computerized. The members of C.I.S. include the Oregon State Board of Education, Oregon Employment Division, Community Colleges, Intermediate Education Districts, local school districts, and both state universities.

The system provides the labor market information to the individual in a number of ways. A questionnaire is one of the main features of the system. This questionnaire is available in a computer program or in needle sort deck cards. The questionnaire helps the students explore occupations related to their self-assessed interests and abilities.

Another feature of the system is the occupational descriptions. This is a brief summary of each occupation in the system, including current local, state and national labor market information. Other features of the system include educational and training information, bibliography and books, cassettes, and names of local people available to discuss their occupations with interested individuals.

C.I.S. is currently available in most high schools and junior high schools in Lane, Coos, and Curry Counties. C.I.S. is also available in some of the schools in Multnomah County. It is projected that next year C.I.S. will be used by 100,000 students in Oregon. C.I.S. is in the process of setting up its system in Jackson and Josephine Counties.

For complete details about C.I.S., write to Bruce McKinlay, Director, Career Information System, 247 Hendricks Hall, University of Oregon, Eugene, Oregon 97403.

Although it is doubtful, in fact not even desirable, that the computer will replace the teacher, the administrator or the councelor, it may certainly change the roles of all educational personnel. Thus, all educators may be able to allocate their time and resources to more effective communication with one another and certainly to more interaction with students in our classrooms and schools. Specifically, the computer may allow the counselor to make better use of his listening, empathetic and re lective skills.

The Future

In any area of endeavor involving the computer, we are limited by our own imagination and creativity. Is it possible that in the future:

1. Every administrative decision will be computer-assisted, and that many decisions will be totally computer made?
2. Administrators will be supplied with a school information packet that contains the accumulated knowledge of many regarding learning, teaching, and administration?
3. An administrator will be able to provide data to a computer as he does now to a secretary or dictaphone and receive output consisting of alternatives with cost/benefit analyses and a recommended decision?
4. Computer systems will be able to output warnings on potential problem areas such as staff conflicts, material shortages, student underachievement, curriculum problems, evaluative crisis on innovative programs?

These questions and many similar ones have already been partially answered by researchers. Perhaps the most ambitious project concerning computer simulation in long-range planning of complex educational systems is the Asian Educational Model (see bibliography). This model was developed in 1966 by two staff members of UNESCO working with five Asian consultants at the UNESCO Regional Education Office in Bangkok. The function of the model is to forecast and simulate educational systems at any given point in the near or distant future. The model serves as a tool to be used by educational planners to demonstrate instantly the implications of any quantifiable change in the educational system or in any factor directly affecting it. Examples include the implications of such decisions as: introducing compulsory education, changing the pupil-teacher ratio, altering the level of teachers' qualifications or their salaries, changing a government's manpower needs or revising the amount of gross national product devoted to education.

A total of 280 questions covering the most important elements of Asian educational systems were framed in algebraic equations and fed into the computer. These questions concerned different educational patterns and specific situations at all levels, in each year until 1989. Given the necessary data (which admittedly are projections in many long-term forecasts) it is possible to get a picture reflecting in numbers what the educational system will be like at any level, at any given date, and in any type of grade or school. It can be learned what school enrollments are likely to be at different levels in any future date, given certain conditions. If a country's target is, for example, 100,000 pupils graduating from secondary schools, the model can tell educators what will be needed on a quantitative basis to bring this about--teachers, primary school intake, capital investment in school buildings, equipment, etc. The model predicts various situations on the basis of given educational hypotheses, and it can also show what must be done to alter these hypotheses, if necessary, to achieve a desired result and situation.

An example:

> A country may, for example, wish to know now what the size of enrollment in the third year of a university science department will be in 1978. The answer, as worked out in less than a minute, would have to take into account such factors as: the number of children born in 1958 and who survived till the age of six when they presumably entered first grade of primary school; the proportion of all six year olds entering first grade in 1964; the proportion continuing through primary school from grade to grade; the proportion entering secondary school and their distribution among various types of specialized education; the proportion continuing on from grade to grade in secondary education; the proportion entering higher education and their distribution among various branches of study; and the proportion continuing through higher education until 1978 (p. 122).

However, in considering such a problem, the determining proportion mentioned above are in themselves influenced by such educational factors as the availability of schools and the efficiency of the educational system itself, as well as by changing demographic situations, social, political and economic requirements and constraints. The UNESCO team believes that the model can be applied to educational systems in all areas of the world at a considerable saving of time, labor, and cost.

A Final Note

The emphasis throughout this article has been possible administrative uses of computers in education. We have not discussed the role of the school administrator as the instructional leader in his school. However, the school administrator is generally the key figure in determining the nature and extent of any major instructional (curricular) change. Thus if computers are to have a significant impact upon instruction it is imperative that school administrators learn about the possibilities, the implications, and the problems. The modern school administrator need not be an expert in the field of computers in education--but it is highly desirable that he be more than a novice.

Resources

Agencies, councils, and organizations available to the administrator for information and assistance in Oregon.

CASEA - Center for the Advanced Study of Educational Administration, University of Oregon, Eugene

CIS - Career Information System, University of Oregon, Eugene

ERIC - Educational Research Information Center, (Administration) University of Oregon, Eugene

NWREL - Northwest Regional Educational Laboratory, Portland

OAEDS - Oregon Association of Educational Data Systems

OCCE - Oregon Council for Computer Education
Dr. Tim Kelly, President
Southern Oregon College, Ashland

OMEC - Oregon Mathematics Education Council
325 13th Ave. N.E., Unit 203
Salem, Oregon 97301

OTIS - Oregon Total Information Service, Eugene

In addition to the above resources, information can be obtained from:

Intermediate Education Districts
State Department of Education
Deans, Schools of Education
University and College Computing Centers
State and national associations of principals, superintendents

Selected Bibliography

General Administrative Uses

Alcorn, Bruce K. "Educational Specifications vs. Systems: Does the Tail Wag the Dog?", Journal of Educational Data Processing; Vol. 2, Winter 1964-65. pp. 99-102

Allen, Dwight; Chatterton, Lynne and Oakford, Robert. "School Scheduling, Practive and Theory," Journal of Educational Data Processing; Vol. 4, Winter, 1966-67. pp. 16-50.

Alspaugh, J.W. "Utilization of Computing and Data Processing in Education," Clearinghouse , Vol. 43, April, 1969.

Ball, Arzell L. "A Superintendent's View of a Computer System," School Management; Vol. 15, No. 12.

Business Weekly. "The Perils of Data Systems," June 5, 1971. p. 62.

Banach, William. "Public Relations, Computers and Election Planning," School Management, October, 1971. pp. 24-25.

Bennett, Lowry M. "OTIS: This Oregon Total Information System," Journal of Educational Data Processing, Winter, 1968-69. pp. 157-165.

Cross, R. and Senderburg, C.H. "Computer Assisted Enrollment Projection Procedures," Journal of Educational Data Processing, Winter, 1967-68. pp. 160-165.

Crowe, Robert L. "The Computer and Personnel Selection," School Management, Vol. 16, No. 8.

Dwarkin, F.W. "Beware of False Gods," Educational Digest, Vol. 36, January 1971, pp. 27-29.

Gauthier, J.R. "Can a Computer Evaluate?", The School Review, August, 1971. pp. 602-613.

Hansen, R.P. "Computers in Education", Clearinghouse , December, 1970, Vol. 45.

Hoag, Ed. "Total Information for Oregon's Schools," School Management, Vol. 17, No. 2. p. 37.

Hogan, D.F. "Improving the Education of Migrant Children," American Education, Vol. 9, April, 1973. pp. 20-24

Maloney, Jr., S.P. "Electronic Data Processing in Education," Journal of Educational Data Processing, Winter 1968-69.

Parker, Tardell and Cerone, James F. "Using a Computer to Project Enrollment Changes," School Management, Vol. 15, No. 10. p. 14.

Saturday Review of Education. "The TIES Network," Vol. 1, May 1973.

Shaw, D.C. "Use of the Computer in Educational Management," Educational Technolog , Vol. 12. pp. 39-44.

Smith, Robin C. "Factors Affecting Administrative Computer Applications," AEDS Monitor, VII. November, 1972. pp. 13.

Whitlock, James W. "Automatic Data Processing in Education," Macmillan Company, 1964.

Winters, William. "What's Involved in the Educational Information System?", Journal of Educational Data Processing, Vol. 5, Winter, 1967-68. pp. 109-115.

PPBS

Handy, J.W. The Computer in Education. Educational Service Bureau, Inc.: Washington, D.C.

Hartley, Jarry J. Educational Planning-Programming-Budgeting: A Systems Approach. Englewood Cliffs: Prentice-Hall, Inc., 1968. p. 76

National Association of Secondary School Principals. The Principal and PPBS. #366 (entire issue), October 1972.

Piele, Philip K. Planning-Programming-Budgeting Systems. Educational Resources Information Center (ERIC), January, 1972. p. 1.

Available Through ERIC:

California State Department of Education. Conceptual Design for a Planning, Programming, Budgeting System for California School Districts, Preliminary. Sacramento: Advisory Commission on School District Budgeting and Accounting, 1969. p. 68. ED 036 124 MF $0.65 HC not available from EDRS. (A more recent publication on this subject is available from the California State Department of Education, Advisory Commission on School District Budgeting and Accounting, Sacramento, California 95814).

Center for Vocational and Technical Education. Program, Planning, Budgeting Systems for Educators. Vol I: An Instructional Outline. Leadership Series No. 18. Columbus: Ohio State University, 1969. p. 279. ED 032 417 MF $0.65 HC $9.87. (Also available from The Center for Vocational and Technical Education, Ohio State University, 1900 Kenny Road, Columbus, Ohio 43210, $4.00).

Eidell, Terry L., and Nagle, John M. PPBS and Data-Based Educational Planning. Eugene: Center for Advanced Study of Educational Administration, University of Oregon, 19 . p. 63. ED 038 741 MF $0.65 HC $3.29.

Katzenbach, Edward L. Planning Programming Budgeting Systems: PPBS and Education. Cambridge Massachusetts: New England School Development Council, 1968. p. 17. ED 025 856 MF $0.65 HC $3.29 (Also available from the New England School Development Council, 220 Alewife Brook Parkway, Cambridge, Massachusetts 02138, $1.00 members, $2.00 non-members).

Keller, Harry. "Development and History of the Concept of PPB." Paper presented at Institute on Program Planning and Budgeting Systems for Libraries, Wayne State University, Detroit, Spring, 1968. Detroit: Department of Library Science, Wayne State University. p. 22. ED 045 113 MF $0.65 HC $3.29.

Piele, Philip D., and Bunting, David G. Program Budgeting and the School Administrator: A Review of Dissertations and Annotated Bibliography. Review Series Number Two. Eugene: ERIC Clearinghouse on Educational Administration, University of Oregon, 1969. p. 50. ED 035 372 MF $0.65 HC $3.29

Scurman, James. "PPBS: Focus on Output Performance. One Local District." Paper presented at annual meeting of the American Educational Research Association, New York, February, 1971. p. 27 ED 047 372 MF $0.65 HC $3.29.

Western New York School Development Council. Development of an Operational Model for the Application of Planning-Programming-Budgeting Systems in Local School District. The Conceptual Bases for the Development of an Administrative Planning System for Local School Districts. Program Budgeting Note 4. Williamsville: 1969. p. 17. ED 039 630 MF $0.65 HC $3.29.

PERT

Federal Electric Corporation. "A Programmed Introduction to PERT." John Wiley & Sons. 1965.

Miller, Robert W. "How to Plan and Control With PERT." Harvard Business Review, March-April, 1962. p. 93

RRPM

Koenig, [illegible]; Kenney, M., and Zemach, R.A. A Systems Model for Management, Planning, and Resource Allocation in Institutions of Higher Education. Final Report. National Science Foundation Project c-518; Michigan State University, 1968.

Western Interstate Commission for Higher Education, Boulder, Colorado:

Clark, David and Huff, Robert. Instructional Program Budgeting in Higher Education. 1972.

Clark, David; Huff, Robert; Haight, Michael and Collard, William. Introduction to the Resource Requirements Prediction Model 1.6. 1973.

Gulko, Warren W. Program Classification Structure. 1972.

Gulko, Warren W. and Hussain, K.M. A Resource Requirements Prediction Model (RRPM-1): An Introduction to the Model. 1971.

Haight, Michael and Manning, Charles. Induced Course Load Matrix Generator: System Documentation. 1972.

Huff, Robert A. and Manning, Charles. Higher Education Planning and Management Systems: A Brief Explanation. 1972.

Hussain, K.M. A Resource Requirements Prediction Model (RRPM-1): Guide for the Project Manager. 1971.

Ziemer, Gordon; Young, Michael and Topping, James. Cost Finding Principals and Procedures. 1971.

Counseling

Loughary, John W. Man-Machine Systems in Education. New York and London: Harper and Row Publishers, 1966. pp. 168-185.

Pierce, Robin. "Careers on the Computer," American Education, August-September, 1972. pp. 31-35.

U.S. Department of Health, Education and Welfare. Computer-based Vocational Guidance System. Summary of papers presented at the Fourth Symposium for systems under development for vocational guidance. Washington, D.C.; U.S. Government Printing Office, 1969. Superintendent of Documents Catalog No. FS 5.225:25053. p. 168.

Weizenbaum, Joseph. "Computational Linguistics," Communications of the ACM, Vol. 9, No. 1, January, 1966. pp. 36-43.

General

Bushnell, Don D. and Dwight W. Allen (editors). The Computer in American Education. New York, John Wiley & Sons. 1967. p. 300.

"Computer to Aid Long-Range Planning of School Systems," School and Society, 94(2274):122, March 5, 1966.

Gerard, Ralph W. (Editor). Computers and Education. New York: McGraw-Hill Book Co., 1967.

Goodlad, John; O'Toole, John and Tyler, Louise. Computers and Information Systems in Education. Harcourt, Brace and World, Inc. 1966. p. 152.

Kaimann, Richard A. and Marker, Robert W. (Editors). Educational Data Processing: New Dimensions and Prospects. Boston: Houghton Mifflin Co., 1967. p. 326.

National Association of Secondary School Principals. "The Computer in Education," #343 (entire issue), February, 1970.

SECTION B: ADMINISTRATIVE WORKSHOPS

by Mike Dunlap

Introduction

Introducing computers into school administration or school curriculum represents a change from the status quo. The impetus for such a change can come from above (the state, school board, superintendent) or below (parents, teachers, students). The school administrator is caught in between and is often the key individual in deciding what will be done and/or the success of the activity.

Teachers and administrators in Oregon who are already involved in use of computers in education have often discussed among themselves what it takes to get wider acceptance and usage of computers. They generally come to the conclusion that the school administrator is the key individual. Thus there has been considerable support in the past few years for efforts to organize and present computing workshops to school administrators. The author of this article has been involved in several of these workshops, two of which were during the late summer of 1973. The ideas presented here proved useful in such activities and led to successful workshops.

A "workshop" might be a few hours in length, or several days. The length of time available will have a considerable effect upon the nature and goals of the workshop. The outline given later in this article served as the basis for a successful three hour segment of a workshop that ran for several days (the remainder of the workshop did not involve the computer field).

In the next few paragraphs we will present eight steps that one should consider following when planning and presenting a workshop. These tend to hold no matter what the subject matter or audience.

EIGHT STEPS TO A SUCCESSFUL WORKSHOP

As one plans a workshop for administrators, he must keep in mind the importance of his task. The administrator needs to become familiar with the use and application of the computer in the school without being turned off. The workshop should strike a balance between exciting the administrator and overstimulating him. It should concentrate on those areas which are of greatest importance to the administrator--curriculum and finance.

The following eight steps are useful if you are planning a half-day workshop or a three day extravaganza. The key point in the process is PLANNING!

1. Determine who the audience actually is. What is the back-[illegible]d of the attendees? Do they have computer facilities?

Are they hostile or do they accept the use of computers in education? Is their major interest curriculum or budget?

2. Commit to paper your goals and objectives. If one can't provide a specific set of desired outcomes for a workshop, then maybe there is no purpose to the workshop. Ask yourself: Why is the workshop being held? What are administrators expected to know when they are done? Do we want to change attitudes? What attitudes? Be certain that the the goals and objectives are on paper so they can be referred to and updated as planning continues.

3. Determine what specific topics need to be presented. The material actually presented at a workshop should be designed to meet the goals and objectives. Each topic should contribute to the overall impression you want to leave in the minds of the administrators.

4. Determine the most logical and appropriate schedule of presentation. Be certain that in the presentation that ideas flow in a logical order, that major points are built out of minor points, and that the pacing of the material is appropriate to the audience. For example, be certain that there is not too much or too little demonstration. Remember that short activites with variation work better than long-winded presentations.

5. Before the session or presentation begins, get to know your audience. Ask each person, if possible, to give his name, tell of his special interest, where he is from, and what his job is. This acitivity accomplishes two goals. First, it permits the administrator to say something about himself and therefore he feels a bit more comfortable. Second, it permits you to see how well your audience matches the projected audience. At this time you must be ready to make last-minute changes in the program that best fit the needs of the audience. If you were expecting mostly principals and you end up with mostly counselors, then you might want to emphasize the use of computers in counseling more than originally planned, and de-emphasize some other topics.

6. The presentation: The presentation is the most visible part of the entire workshop. Even if the other planning has been done carefully, it is still possible to fail here. Here are two important suggestions:

a) Get the very best speakers to deliver your talks; ones who are the most interesting and have the widest backgrounds.
b) Be certain that the mode of presentation is appropriate to the group. In the metropolitan Portland area administrators might like a very formal presentation; but in eastern Oregon they might appreciate a down home chat.

7. Evaluate the workshop. This consideration is linked to the development of goals and objectives. You know that you ac-

complished something if you take time to evaluate. It is important to consider the evaluatuon at the same time you develop the presentation. Whatever the form of evaluation, it should be specified in writing.

8. Document what you did. Administrative workshops are becoming more and more important and frequent. Document your work and make it available to others so that every workshop does not result in a re-invention of the wheel. Other people need to know what you did; share with them!

A PROFILE OF A SUCCESSFUL WORKSHOP

The workshop described here was designed for a group of about twenty administrators from small schools in eastern Oregon. None of the schools had computing resources. The time allocated to the workshop was three hours, and was part of an in-service program which lasted several days. Two people, with extensive backgrounds in instructional computing, were selected to present the workshop in conjunction with leaders in computing from the local college.

Planning

Step 1: The two speakers allocated three hours of planning time for a general discussion of administrator workshops and the needs of this workshop.

Step 2: A set of outcomes was developed:

1. Develop awareness of available instructional materials and support facilities for instructional use of computers.
2. Assist in development of "literacy" about the computer and its use in the educational environment.
3. Provide a rationale for instructional computing.
4. Convince the administrator that he is important in the development of instructional computing.
5. Provide some suggested steps to bring about the desired change.

Step 3: Planning for the workshop.

A listing of suitable topics was constructed that supported the five major outcomes. These topics were then evolved into a presentation sequence which resulted in three presentation periods alternated with three demonstration periods, each of which was limited to about thirty minutes.

Step 4: Development of evaluation forms.
Step 5: The entire plan was submitted to two independent "experts" for their comments and suggestions. Some revision was made to the plan at this time.

Presentation

Session 1: The fist session began by asking each participant to give his or her name and position. The speakers were then able to determine that the audience was the expected one. The next 30 minutes were spent in considering some basic information:

a) Terminology--the terminology necessary to understand computers in instruction was explained, defined, and illustrated. (Timesharing, batch processing, teletype, terminal, simulation, etc.)
b) A profile of a successful project. The Rogue Valley Council for Computer Education was discussed in detail. The talk focused in the development, organization, progress, funding, and support of the RVCCE. This presentation was designed to show that others, in far parts of the state, had been successful in obtaining instructional computing through the combined efforts of a group. The strong suggestion was made that this could be done again.

Activity 1: After 30 minutes of presentation it was time to break and change the pace of the workshop. This periodic break maintained interest at an extremely high level.

a) A handout was given each workshop participant. It contained the output of several instructionally oriented programs. One was presented from each of the following areas: counseling, CAI, simulation. Each program was explained, and the participants allowed to select one to be demonstrated.

b) The participants then went to the demonstration area and began a run of the Huntington water pollution study--POLUT (this is the topic they decided would most interest them).
After a few minutes the administrators were running the terminal themselves. Throughout the remaining activity periods the administrators ran the terminal and no significant technical assistance was needed. This ability to run the machine themselves, and their ability to call up and run programs, helped to underscore how easy the computer is to use. (Even though we were about as far from the computer as we could be and still remain in Oregon, Ma Bell performed flawlessly in maintaining our long distance connection.)

The participants enjoyed the time with the computer and the ability to move about the room. The freedom of movement provided an opportunity to talk one to one with speakers. Little discussion groups were formed, and many of the outcomes of the conference were carried out informally.

Session 2: This session was devoted to the instructional uses of the computer. It began by discussing the major areas in instructional computing: computer literacy, CAI, CMI, CAL(CEI), programming and occupational counseling.

a. Each of the major ideas in instructional computing was defined, explained, and discussed. With each idea the speaker presented one or two examples of its actual use in the school and the benefit it could bring to the instructional program.

b. One of the major points stressed was that a school with a small budget could simulate, on a computer, many of the experiences that were too expensive to bring directly to the students.

c. The question and answer period following the presentation showed that the administrators desired these facilities for their schools.

Activity 2: The previous 30 minutes of presentation left the administrators anxious to try out some of the programs mentioned. They seemed to have a special interest in the counseling ability of the computer and spent much of the activity period exploring with the aid of the Occupational Information Access System.

Session 3: This was the wrap up session. It attempted to tie together all the information that had been discussed in the small groups with the information presented in the formal sessions. One of the key points concerned ways that this group could get started in instructional computing.

a. Among the major ideas for starting an instructional program was the need for mutual cooperation.

b. Sources of fu ing were considered, with suggestions of where they would purchase computer facilities or computer time. The power of their cooperative ability to obtain finances was discussed along with the resources available through their local Intermediate Education District.

Activity 3: The last activity session began with the evaluation of the workshop. The participants were pleased to help us evaluate the result of the experience. The participants were formally excused and thanked for their participation. However, many stayed to either talk in small informal groups or to run additional programs on the computer terminal.

Evaluation

While the workshop did not fully meet all of its desired outcomes, the administrators did: develop an awareness of available instructional materials and support facilities; they were somewhat more literate about the use of computers in instruction; many more clearly understood the importance of the administrator's role in instructional computing; and they had several workable suggestions as to how they could get started.

In general, the workshop was a success. The ultimate success of such an endeavor is to watch and see what activities happen in the future.

SUMMARY

In this article we have discussed the importance of administrator workshops. Included was the idea that the administrator is a key person in the instructional application of the computer. Eight steps were presented which lead to the development of a successful workshop. Finally, a profile of a successful workshop was given. That workshop was developed using the eight steps outlined in this article.

The importance of administrator workshops cannot be discounted. The key idea in the development and execution of such workshops is PLANNING!

Section C: Abstracts of related articles for chapter VII

Banach, William J.
Public Relations, Computers and Election Planning
School Management, October, 1971 Pages 24-25

What does it take to pass a school financial measure and how can we predict the outcome of the election? This issue is of vital importance to school districts throughout the nation, if we are to maintain public schools as a functional institution in our society. By integrating good public relations and computer science, the Rochester (Michigan) Community Schools have developed a plan of action that increases the chances of favorable election results and involves the community in school support programs. This model could provide insights for other school districts preparing to go to the polls.

Public relations experts recognize that there are many different "publics" in a school community. The computer provides us with a tool to identify and isolate each "public" rapidly for better election planning. The information found in voter registration records is key punched into data cards: names, address, city, sex, ages, elementary attendance area, precinct and voting frequency of every registered voter in the district. The community is sorted into "yes" and "no" groups, and the computer printouts clearly indicate the number of "yes" voters in the district. With this information a campaign can be geared to get the necessary number of "yes" voters to the polls.

In the Rochester district, pre-election data indicated that major support would come from three specific groups: 1) preschool and elementary parents; 2) district employees; 3) educators living in the district but working outside the district. Twelve hundred people in the preschool and elementary public were registered but had not voted in any school election in the past five years. Special campaign literature was developed for this particular group and an all-out effort was made to get all the non-registered voters in this group registered. After the registration deadline passed, the computer was again used to index and up-date the information on each voter for future retrieval and analysis. From these printouts came the information that led to the organization of election groups to insure face to face contacts with potential "yes" voters.

On election day, poll watchers were furnished an alphabetical "yes" file by precinct. As voters appeared, their names were checked off if they were on the "yes" file; otherwise, they were counted as a "no" vote. The results were called in hourly and a comparison was made against previous voting frequencies. This gave an up-to-date analysis so that last minute strategies could be employed. The election turned out a success. With the almost unlimited capabilities of the computer in analysis of data the administrator can plan a campaign directed to the people most likely to pass a school issue. The community is willing to work harder when such information is available and there is some assurance of success; and the success itself, tends to bring about better public relations.

Paul S. Ashdown

Dorkin, Edward Paul
Beware of False Gods
The Educational Digest, (January 1971),
Pages 27-29

The title of this article states the position of the author. Technology has entered the realm of the behavioral sciences, and guidance personnel have accepted it, or as the author feels, overaccepted it. He feels that counselors should approach this new technology with a great degree of caution. Man's preoccupation with technology has led to many problems. We have become far too materialistic. The USOE subcommittee report on easing tensions (1969) said that technology was either directly or indirectly the cause of most conflicts between schools and students. The report further stated that the most common reasons cited by students for causing conflicts were (1) dehumanization of society, (2) inequitable distribution of wealth and power, (3) irrelevance of education and (4) cultural exclusion. The author feels that guidance personnel should work toward solving these problems.

To solve these and most other problems, some guidance people feel that all they have to do is to get their hands on a computer. The author wonders if it is proper to use technology--the thing he feels directly contributed to the formation of these problems--in the solving of these problems. He is afraid that too much hope is being placed in technology and therefore, not enough human effort is going into the solutions.

This, he says, does not mean that guidance technology should be eliminated, but rather that it should be considered as only one tool to help with guidance problems. Therefore, it appears that the question is not whether or not to use guidance technology, but how to best use it.

He lists the following, which he feels are limitations of guidance technology:

1. Computers can provide speedy retrieval and manipulation of data, but cannot deal with complex human problems.
2. Computer hardware (the machines) are now in a very advanced stage, but the software (the programs), with respect to their applications, are still in a very beginning stage. Adequate software for guidance has just not been developed.
3. There are many interesting packages (canned programs) provided by different manufacturers, but as long as these packages are being prepared by businesses whose only motive is profit, users of these packages are not getting the best packages for their individual situations.
4. Most counselors lack adequate training in how to use these technological innovations. Because of this lack of training, the technology available is being abused.

In summary, the author does not feel that the technology available should not be used, but rather, we should be cautious when we use it, and not use it just because it is available. He also feels that the technology is ahead of our understanding of how to use it. Of course, if this technology is not handled properly, it is the student who suffers.

John Shirey

Hemphill, H. David
THE PROMISED LAND OF THE COMPUTER
EDUCATION CANADA, Vol. II, n. 2, pp. 47-50

"The rapid growth of EDP...is having, and will continue to have, a great impact on all those associated with education." Contemporary technology has very high potential for the improvement of education, but educational organizations have been slow to adopt the equipment and techniques of our technological age. Because of its versality, the computer has a very high utilization potential.

This article is adapted from a paper presented at a conference on computers held at York University, in 1970. It is divided into two major dimensions:

(1) The System Elements: The major input elements considered were equipment and personnel. The major output elements were operations or service provided, divided as: (a) Business management, (b) Educational management, (c) Instructional management, and (d) other services not included in the other three.

(2) Stages of Development: (a) Planning, (b) Developing, and (c) Operating

Eight of ten provinces participated in the survey. "The tabulation...showed that the greatest increase has come in small (fewer than 10,000 pupils) and medium (10,001 to 50,000 pupils) sited units. ...The number of districts using EDP has increased from 14 (1966) to 83." The fact that the growth is exponential is partially due to the availability of payroll and voucher accounting packages. Almost all districts of 25,000 or larger are using EDP in some form.

By looking at the data on operating, developing and planned output of EDP is far greater than that currently being used or being developed. To bring this about, an increase of 50% or more in personnel is anticipated. And, equipment increase will be even larger. Also there are 56 computers now in use, with an additional 47 planned for future operation. At this time, most of the development is taking place within the provincial departments, rather than at the local unit level.

MAIN IMPLICATIONS

(1) Provincial departments of education must prepare for an increased call on their EDP services from local administrative units.

(2) The often-called-for provincial leadership in EDP seems to be emerging in the form of integrated information systems.

(3) Perhaps the need for educating educators in the characteristics and potential of EDP in their profession is of even greater importance.

(4) Almost all of the current, developing, and planned use of the computer in Canadian education relates to administrative applications.

Gerald Larer

Hill, L.J.; Hughes, J.H.; Tennyson, J.W.; Epting, L.B.
Computer-Aided Operation of Cooperative Education
Engineering Education, ASEE, Vol. 61, No. 7, (April 71), pp. 803-805

This article describes the application of the computer to handle the needs of the cooperative education program at the Mississippi State University. The system was used to handle record maintenance and provide a rapid-access information retrieval function to aid day-by-day program operation.

Inputs to the data system originate with students, employers, university, and the co-op office. Outputs from the system consist of periodic reports and day to day records requested by the co-op office. The basic information contained in the system consists of 30 listings which serve both the internal management demands of the co-op program office and as external academic reviews.

Operational aids are generated within the system to provide such functions as categorizing students and employers, generating a resume for each potential co-op student, and checking evaluation forms of students. It has also provided a means of evaluating the methodology of operating the co-op program at the university.

The co-op office is continually refining the system and plans on adding a computer-aided co-op student placement function to the system. An attempt to analytically review the cooperation education program in general has been initiated.

Robert Layton

[illegible]ay, Ed
'Total Information' For Oregon's Schools
School Management, Vol. 17, No. 2, (February 1973)
Pages 37-39

OTIS-the Oregon Total Information System-was started in the mid 1960's by a group of Oregon school superintendents, under the direction of Dr. William P. Jones, the Lane County IED Superintendent. Financial support was obtained from the U.S. Office of Education. It is a cooperative computer center offering a range of data processing services to educational institutions. Actual operation began in May 1968. Steady growth has expanded the services to 250 schools with more than 100,000 students in 50 school districts, and several community colleges and educational agencies in Oregon. Currently, there are 119 terminals on-line to the three-computer data processing configuration. OTIS is being used for three basic kinds of services: 1) student services, including computerized enrollment, attendance accounting, mark reporting, student class scheduling, test scoring and health survey reports, 2) business services, including fiscal accounting, payroll, personnel and inventory, and 3) instructional services, including problem-solving, CAI, the Occupational Information Access System for student vocational planning, and the OTIS Automated Library System. As important and comprehensive as these services are, OTIS does more: it provides educational administrators with the ability to retrieve from their data base vital information necessary for decision-making and planning.

The OTIS system is available 10 hours daily and routinely handles 3,500 inputs an hour. The OTIS configuration is composed of a Tempo 270 T terminal control processor, an IBM System/360 Model 50 and a Hewlett Packard 2000F System.

The organization currently operates on receipts from member school districts. The basic charge for administrative services is $8.80 per student per year. Four times the volume of services with lower expenses and smaller staff are now provided in comparison with the first year of operation.

Although not a state or county agency, OTIS is organized under the legal authority of the Lane County IED. Guidance is provided by an advisory board made up of educators from throughout the state.

Administrative services have made up the bulk of the volume of OTIS activity, but in the past year there has been a decided increase in problem-solving. Indications are that this area of student learning will continue to increase. OTIS recently acquired a library of programs for computer guided student instruction in several academic subject areas.

OTIS was born with two broad goals: to build and maintain a service organization that could provide Oregon schools with the hardware and technical staff necessary to meet their requirements, and to develop with each user a plan that would assure individual satisfaction as well as efficient and economical utilization of the OTIS hardware and staff. OTIS has largely met these goals and objectives and is now in the process of determining the direction of its future growth.

Hoerger

Hogan, P. F.
Improving the Education of Migrant Children: MSRTS
American Education Vol. 9 (April 1973)
Pages 20-24

Since November of 1968, a computerized communications network has been in effect among 137 terminals along the paths of migrant workers in 48 states. The network can, in about four hours, provide a school with the basic academic and health history of any of the 390,000 migrant children registered with it. Using Title I funds from the states, the Arkansas Department of Education, which now has its own computer for MSRTS, can take information fed into the central data bank by teletypes in one school district, and, upon a proper query, can send it to the district where the child next attends school.

Information thus transferred includes the child's name, sex, birth date and place, I.D. number, innoculation and vaccination shots received. Also, it tells the diseases he/she has had, the results of a complete medical examination, the results of aptitude and progress tests, and special abilities in the arts and scicnces. Teachers rate the child's academic record under 39 headings and sub-headings covering the basic school subjects, especially language skills, but nowhere can they write their own comments, thus supposedly protecting the child's right to privacy.

Critics note that this is still a data bank of personal information which can easily be misused. Arkansas officials have already limited access to school authorities in migrant programs, and restricted the kinds of information in the child's file. Yet some teachers feel the form they fill out is too restrictive. Another major problem is the failure of some school districts to send the child's progress report to Arkansas immeciately after the child leaves their school. The child's next teacher is then unable to effectively continue the child's education right away; this is critical, since the teacher may have onlv a few weeks to do anything before the child's family moves on.

The claim that the system is discriminatory in that it doesn't help all children, is affirmed and expanded by Cassandra Stockburger Director of the National Committee on the Education of Migrant Chi[illegible]ren, who wants the money spent on tracking records to be spent improvi[illegible] he schools themselves. MSRTS agrees that improvements are necess[illegible] and hopes that MSRTS can be "a significant move toward building a found[illegible]ion for a co-ordinated, broad-scale planning effort that crosses state and district lines."

Marlyn J. Kern

Hedges, William D.
The Principal, the Computer, and Emerging Applications to Instruction
Phi Delta Kappan V. 55, n 3, (November 1973) pp. 174-176

For administrators who received their graduate training 20 years ago, the computer was unavailable. Today the computer is available but few administrators are receiving training in understanding or using the digital electronic computer. For these persons, the author briefly describes the available and emerging instructional applications of the computer; suggests minimal skills and background the interested principal or curriculum director might well develop; and describes practical first steps feasible in many schools at this time at elementary through senior high levels.

Under the first topic, it is suggested that the computer can facilitate instruction in six major areas: 1) drill and practice, 2) simple games, 3) problem solving, 4) inquiry mode, 5) individualized testing, and 6) calculation, and mentions other emerging areas of usage in common management functions such as scheduling, payroll and data analysis.

Next, it is suggested that the educator take a course in one programming language such as BASIC, FORTRAN, PL/1, APL or COBOL designed for the professional educator. He should also begin reading technically oriented journals to familiarize himself with the hardware and software available in order to be better able to make decisions concerning its use in his school system.

Practical first steps suggested are to persuade the school board to rent a single computer terminal for one academic year with a faculty member from the high school supervising its usage, since terminals are the least expensive method of entering the computer field without a huge outlay of funds. It is further recommended that the educator get his faculty and students busy working with the computer and in this way introduce instructional applications with a very modest investment.

Doris J. Nelson

WILBUR, Thomas P.
A Rationale for the Employment of Systems Analysis as an Aid in the Management of Public Schools
Journal of Educational Data Processing, Vol. 5, No. 1, Winter, 1967-68
Pages 5-12

This article generally sets out a model for a systems approach to educational administration, states some obstacles retarding wide use of a systems approach in school management, and cites several forces which, the author believes, will cause the systems approach to become widely used. The model views education as a system composed of inputs, processes, and outputs.

Wilbur suggests that there are three steps in the model: its formulation, with the situation of any particular school district or school plugged in; the use of the model in making administrative decisions every day, and the evaluation and re-cycling of the decisions and their resultant programs.

The article then states four problems related to the current lack of widespread uses of this systems model.

1. Lack of performance criteria and difficulty in gathering data.
2. Time, manpower, and equipment costs.
3. The nature of educators.
4. The uncertainties of systems analysis.

Despite these problems, there are three compelling reasons that will cause the gradual adoption of systems approaches in a majority of the nation's schools. These reasons are:

1. The need for methodical, analytic problem solving.
2. The rapid recognition of impending problems.
3. Spin-off and stimulus to innovation.

The author concludes that this systems approach is an aid, not a panacea for the school administrator, but that it will prove an invaluable tool.

Ken Bierly

Winters, William K.
What's Involved in This Educational Information System?
Journal of Educational Data Processing, Vol. 5, No. 1, Winter 1967-68
Pages 109-115

If an educational information system is going to be implemented in a school system, then there are three problems that need to be dealt with, 1) deciding what is to go into an educational information system file; 2) the interrelationship of data files at the same level and at different levels; 3) and a file management language.

To classify data in determining its usefulness, the following three divisions can be made. a) Items that are absolutely necessary; i.e., staff names and their social security numbers. b) Data items that have actual or anticipated future educational information value. For example, in a university information system student file, the numeric code which represents the elementary school that the student attended might be put into this file. This would provide a means of tracing the student's educational background. The idea to keep in mind is that no information can be output unless data is stored. c) This classification involves data items that have no present or anticipated relevance. Data items such as color of students eyes and hair would be categorized in this group.

When deciding what goes into the file, the educator must be able to communicate to the system analyst the kind of output he expects. The educator should be able to outline the kinds of questions he wants the system to answer. This will help the system analyst to design and furnish the kind of system that will perform the service desired.

Data entered into the system must be put into the proper basis file and keyed to the proper file level. Information is keyed in terms of the individual, school, school district, state, and federal. At each level, the data will be used in a different way. If each system gets its data from a lower level, then each system can benefit from its own information as well as all the systems below it.

A set of programs refered to as GIS (General Information System) makes available to the user, in terms of a query language, a means of defining and storing as well as maintenance, retrieval and output of data. GIS makes it much easier for those interested in setting up a system to experiment with various concepts of file design to fit their particular needs.

Paul S. Ashdown

CHAPTER VIII

COMPUTING FACILITIES

by

Fred Beisse

Cliff Burns

Robert Christensen

Mike Dunlap

Richard Haller

Cliff Houge

Charles Klopfenstein

Bill LeMaster

David Moursund*

Rusty Whitney

*Cliff Burns was group leader for this chapter; he and David Moursund served as editors for this material.

SECTION A: OVERVIEW

A computer system consists of hardware and software. A computer system occupies physical space, and generally is supervised by one or more people. This chapter considers all of these topics.

Many schools (particularly secondary schools and small colleges) do not yet have computer facilities. By and large they do not have personnel who are qualified to carry out the steps which should be followed in the computer acquisition process. The first step is a careful needs assessment. This requires a good understanding of what tasks, administrative and instructional, can make effective use of computers. Next comes the specification of computer hardware and software appropriate to fill this need. By and large these specifications should be independent of any particular computer manufacturer. Then comes a request for bids, and evaluation of the bids. This requires good knowledge of technical aspects of computer hardware and software, and insight into the various computer manufacturer's modes of salesmanship. Two sections of this chapter cover these ideas in more detail.

If one is going to use a time-shared computer or a mini computer he will need a computer terminal. Three common types of terminal devices are hardcopy terminals (like an electric typewriter), cathode ray tube terminals (like a TV set with a keyboard), and card reader terminals (to read mark-sense and punched cards). A rather lengthy section in this chapter discusses the three types of terminals in detail.

The position of director of a medium or large computing center is very demanding, and is generally filled by a professional in the computing field. The small computing facility, as one finds in a secondary school, is usually directed by an amateur. One section of this chapter is devoted to [illegible]sing the duties and responsibilities of the small-computing-facility director. It is recommended that he be given a substantially reduced teaching load and/or extra pay for handling this job.

The question of whether to join in a district, statewide, or national computing network faces many schools today. If a school already has computing facilities it may fear that the quality of services it provides its users will decrease upon joining a network. If a school has no facilities, it will want to insure that it gets the best possible deal if it decides to join a network. Two sections of this chapter discuss computer networks and the components of a good quality computing service.

The overall message of this chapter is plain. If you want to be involved in the computer hardware and software acquisition process you should have a good understanding of the ideas discussed in this chapter. If you don't have this kind of knowledge, bring in a consultant who does!

SECTION B: ACQUIRING A COMPUTER FACILITY

by Robert Christensen and Cliff Houge

More than 100,000 computer systems have been installed in the United States. Thus a large number of people have had experience in going through the computer acquisition process. However, less than 5% of these computers have been installed in educational institutions, and fewer still of these have been installed in the public schools below the college level. This article is primarily directed at teachers and school administrators at the precollege or small college level who might be concerned with the computer acquisition process.

Generally public school teachers and administrators have relatively little knowledge of computers, and no experience in acquiring computing facilities. The first thing to realize is that the process in non-trivial. It is much more difficult than buying a car; rather it is more like having a house built. But in the house building case one has lived in a house for many years, and has been in hundreds of different houses. A house is a familiar, everyday object. Such is not the case for computers.

The house analogy also breaks down in the effects of an error in judgement. A house will provide shelter, although it may be too little or too much. In any case one can usually sell a house quite readily (often at a profit) or add on a couple of rooms. If one gets stuck with an inappropriate computer facility it usually costs considerable money and time to remedy the error.

Before you decide to acquire a computer facility there are several things that should be done. Some of these things will depend on who you are. If you are a teacher you must convince others, particularly the administration and the school board, that a need exists. Even if you are the superintendent or the school board you should be certain that a need exists before you decide to actually acquire anything.

Let us assume that the superintendent and the school board are convinced that some sort of action is called for. This action should take the form of a careful needs assessment, and should be done before any other action is taken in the line of acquisition. You may find that you don't need a computer facility at this time.

Some of the things you should consider in assessing your needs include the following:

1. For what purpose or purposes would you use a computer? Will the expected uses be mostly administrative, such as financial services, educational management, and student record keeping? Or will it be instructional in nature such as computer science, problem solving and CAI?
2. How many people are going to use the computer, and how widely are they dispersed?
3. If the computer is to be used for instruction, what types of input/output devices are to be used?
4. If used for instruction, what subject matters are to be covered, and what is the level of the users?
5. What computer languages are necessary to do the job? This

will be determined by the purpose for which the computer is to be used.

6. What is the desired or necessary turnaround time for programs submitted to the computer system (the amount of time elapsed from when the program is submitted until results are obtained)?
7. What type of peripheral devices (i.e., line printer, card reader, cathode ray tube, etc.) are needed?

In order to properly conduct a needs assessment, select the correct equipment, and make the final selection from bids, you should seek the advice of a consultant who is knowledgeable in these matters. (This should not be a salesman or representative of some computer company.) There are many people available free of charge as consultants. Many colleges or universities can either provide or recommend consultants. Even if you must pay for this service it is well worth the investment. The amount you spend could very well be saved many times over if your needs are more accurately assessed and the final selection is the best one for your situation.

In order to do a proper job of needs assessment you should have have some idea of what is available on the computer market. There are many types of computers that operate in many different ways using many different languages. A brief summary of a few key ideas is given in the following paragraphs.

Computers operate primarily in one of two modes, the batch mode and the interactive mode. In the batch mode the programs are run through one at a time. The computer does one job or "batch" completely before starting on the next one, which is then done completely, etc. In this mode the program is completely coded, frequently on punched cards, prior to being entered into the machine. The turnaround time can vary from a few seconds to several days, depending on the location of the computer and who owns and runs it.

In an interactive mode the person using the computer sits at a keyboard terminal (think of it as an electric typewriter connected to a computer) and interacts with the computer as his program is being processed. It is possible for a number of terminals to be connected to the same computer and to be operating simultaneously. This is called interactive time-sharing. Terminals need not be adjacent to the computer, as they can be connected to the computer via telephone lines or direct wires. In time-shared computing the resources of a computer system are shared among a number of users. A medium size computer can handle 30 to 60 users simultaneously, a small one from 2 to 32, and very large computers can handle several hundred simultaneous users. The response time for each user is usually rapid enough so that very little or no delay is noticed when he is interacting with the computer. The interactive mode is very popular in teaching students how to use the computer. In fact, many schools start using a computer by buying time on a time-share system.

A computer system consists of physical machinery called the hardware, and computer programs called the software. The software is designed to help people use the hardware. These concepts are explained in more detail elsewhere in this Handbook, and they should be kept in mind in making any decisions about what computer system to acquire. In order to do any specified job, both hardware and software are required. Both cost money, and not all computer systems are good in both aspects. A relatively inexpensive piece of hardware might require some very expensive software to make it do a desired job.

Computers come in many sizes, ranging from a mini-computer which may cost as little as $5,000 to a maxi-computer which may cost 10 million dollars. Different machines have different features which affect the cost and also their applicability to a particular situation. Usually the more features there are, the more the system will cost. Features for which one can expect to pay more money include such things as greater storage capacity, more terminals, different types of input-output devices such as line-printers, cathode ray tubes, and card readers, different language handling abilities, faster processing speed, etc.

The needs assessment should result in a statement of what will satisfy your needs in terms of number of terminals, amounts of storage, amounts of batch processing to be done, etc. It is a statement in terms of hardware and software necessary to do the job you want done. It should be independent of any specific vendor's equipment!

Once a decision has been made as to what is needed in the way of hardware and software, one needs to examine the various ways of acquiring these things. These ways depend primarily on your locality. Metropolitan areas are able to provide more variety than remote areas. Some different ways of acquiring computer services may include the following:

1. Service Bureau. A data processing service bureau sells the use of the computer on its premises. This means that if you want to use the computer you must go to the service bureau. Normally this would limit operation to the batch type mode. The turnaround time would depend on your distance from the service bureau and on their operation. This could mean a matter of a few minutes or hours if a messenger took materials and waited for them to be processed, or it could mean several days or even weeks if material was mailed in to be processed. One could possibly get another school with a computer to act as a service bureau.

2. Computer Utility. The computer utility normally installs a teletype or other type of input/output device such as a card reader/line-printer on the user's premises attached via a telephone line to a computer on the utility's premises. This allows the user to interact directly with the computer. It usually allows the user to have access to a larger computer system than a school could afford to own. The cost of using computer utility services has decreased markedly in the last few years. In using a computer utility one can expect to pay for four things: The initial hookup, time while attached to the computer (connect time), actual computer time used (CPU time), and storage charges for files you may wish to store. Frequently there will be some added communication costs such as use of the telephone line. A possible starting point for a school which has limited funds

but wishes to get started in computing is to hook up for a short period of time to a computer utility.

3. Lease or Rent. It is possible to have a complete computer system on your own premises by leasing or renting equipment from some manufacturer or from a third party. The lease may be in the form of a lease-purchase-option whereby you own the equipment offer leasing it for some set number of years. There are certain advantages to having your own system. It is always there and accessible for your use when you want to use it. Moreover, leasing gives one considerable flexibility. If you wish to upgrade or replace equipment, the company from whom you are leasing takes care of this. They also will take care of the maintenance. Be sure that any provisions you wish made for upgrading or replacement are specified in the lease. These must be spelled out in careful detail to protect you.

4. Purchase. Certainly one of the most obvious ways of having equipment on your premises and at your disposal when you want it is to purchase that equipment. With the advent of the mini-computer, the possibility of a school owning its own computer system is a realistic alternative. Sometimes it is possible to find a used system that meets your needs at a price considerably lower than that of a new one.

5. Consortium. Another alternative worth investigating is a consortium, where several school districts go together to set up a data processing center. A consortium allows each school to have access to a more expensive system than any of them could probably have alone.

Now that we have considered the needs assessment and some of the alternatives available for acquiring a computer facility, there are some other considerations before a decision can be made. Probably the first consideration is cost. No matter what your needs, or what you may desire, you can't acquire something if you do not have the funds. If your funds do not allow you to meet all the needs you may have to reassess to find which ones can be put off until a later date. Try not to acquire something you will outgrow in a short time, particularly if you are going to purchase. Many times upgrading a system costs more than it would cost to buy a whole new system that will operate at the desired level.

Consider whether a system requires a special environment in which to operate. Some systems require special electrical circuits and air conditioning which could add considerably to the cost of owning them. Some may require more space than you have available. They may require specially trained personnel to run them. Consider the amount of money required for paper, cards, paper tape, printer rolls, etc. Consider the type and amount of maintenance required. Maintenance costs may run as high as 12% of the original equipment cost, per year.

Before making a final decision on what computer facility to acquire visit a computer installation which uses the type of equipment you are considering. This gives you an opportunity to actually see it in operation and to inquire about how well it does those things you want to do. If a vendor cannot show it to you in your desired configuration you can be pretty certain it does not exist. Don't let a computer manufacturer experiment on you!

This is certainly not an exhaustive list, but it does give you an idea of some of the things to look for. As you become better informed you improve your chances of making a decision which will be best for your situation.

Once a decision has been made as to what is the best system for your needs, you must then go about the procedure of actually acquiring it. Whether you plan to rent, lease, or purchase you should request bids from the appropriate organizations. The more competitive the bidding situation, the better your chances will be for getting the most appropriate computer facilities at the best price.

In requesting bids, you should include the general specifications that are included in any bid received by the school district. Bid specifications should also include some very specific points. You should be allowed time to make a decision; therefore bids should not be allowed to be withdrawn for a period of time after the deadline for bids to be in. If the bids are for lease or purchase, request that bids include all costs necessary to get the system up and running, such as freight, installation, etc. It is probably a good idea to specify a date 90 days prior to your absolute deadline for the system to be up and running. You may wish to make provisions for penalties if this date is not met and you feel you will suffer as a result. Keep in mind that penalty clauses may significantly increase the cost of your contract. Ask that bids specify the configuration, giving the manufacturer's name and model number for each item, the ability to expand to a given size, and the cost of this expansion. Ask that bids specify what peripherals may be used and the cost of installing any of them. Bids should also specify the cost of necessary maintenance and who performs the maintenance. If there is a need for a special environment to operate the system effectively, this should be specified. Request that any user's group or library of programs which may be available for the system be included in the bid. In summary, acquiring a computer facility involves much time and work if you are to get the one system which is right for you. It is very important to get professional help in the selection process. If you are careful and systematic in your selection process you will be rewarded by knowing you have the best system for your situation and for the money you have invested.

BUYERS' GUIDE TO INSTRUCTIONAL COMPUTING

What to Look For
What to Avoid
Hints for Dealing with Vendors

Preparation of this Guide is jointly sponsored by the Oregon Council for Computer Education and the Communication-Resource Center of the Oregon Museum of Science and Industry. However, opinions expressed are the responsibility of the author.

Rusty Whitney
OMSI, Portland, Oregon 97221
(503) 224-9500
May 20, 1974 (Fifth Edition)

Contents | Page

Printed in the U.S.A.

WARNINGS

"IF YOU CAN'T SEE IT, THEN IT DOESN'T EXIST." --- Gunderson's Law

This is the single most important rule. Most grief comes from buying "computer futures" --- systems that almost exist. If you know what you want, ask vendors to transport you at their expense to see customers who have current production models in use with the type of applications you expect. Jim Gunderson of the Multnomah County Intermediate Education District used this rule while advising the committee of teachers which helped set up Oregon's first cooperative instructional computing utility.

EVEN IF YOU DO SEE IT, IT MAY NOT EXIST

Don't settle for a special factory organized dramatic presentation designed for your sole consumption. It's flattering but perhaps misleading. You would not buy a car without testing it under real road conditions; don't buy a computer system without seeing it "under fire."

Remember, they are not doing you a favor --- marketing computer systems can represent a large chunk of the manufacturer's costs and smart vendors expect to spend considerable effort in attracting you to their product.

PREVIOUS CUSTOMERS MAY HAVE A VESTED INTEREST IN THEIR MISTAKES

When visiting a computing system, try not to be a burden to the system manager but try even harder to get facts both for and against the system rather than just good vibrations. A man who bought a $150,000 lemon may not want to admit his mistake, even to himself.

COMPARING SYSTEMS: ABOUT BENCHMARKS

Everyone seems to have her own favorite three line program for testing the speed of a system. Such benchmarks will tell you how well the system runs that particular program --- no more than that.

Vendors all spend a little time on dirty tricks. A favorite is to amaze a customer with the performance of the system on a particular program knowing that the competition's machine will do poorly. The salesperson for even the puniest system can find features that are not present on other systems.

However, if you can define a number of jobs that are for your purposes "typical", you should try to devise methods for making meaningful comparisons of running times on different systems.

GENERAL CONSIDERATIONS

ABOUT THE WORD "SYSTEM"

"System" is more than a fad word. In the past, when ninety per cent of a computer center's budget went to purchase a very obvious and rather cranky computing machine maintained by a very small band of specialists, the computer was the system, or at least most of it. But today, the machinery is not only much more powerful and useful but it is physically smaller and often cheaper than other budget items including operators, data clerks, programmers, maintenance staff, and administrators. The computer hardware could be working fine, but if 455 identical parking tickets are issued to one person everyone says "the computer did it." No. The SYSTEM did it.

<u>This is not a trivial semantics game!</u> Most computing catastrophes are not caused by equipment failure. Really! Inanimate computers make splendid scapegoats for bad planning or inept use but if you are to avoid being the victim of your system, start figuring out what you want your system to do.

ARE BIGGER COMPUTERS ALWAYS BETTER?

Conventional wisdom in the computer industry has long recognized economies of bigness --- if you trade in ten small computers for one big one you can fire nine operators, right? Sure, and besides, although the big computer may cost $1000 per hour to run, it will handle bigger jobs more flexibly. One government financed effort hopes to hook 2000 educational terminals to such a giant giving fifty cent per hour time (exclusive of communications charges).

Unfortunately for conventional wisdom, most big computers have been big flops at providing reliable, economical, responsive computing to the present wants of the educational community. IBM has tried and failed badly; so have others. Not only has the cost per hour of computing been high but big computers require a big support staff which often fails to put instructional computing first.

As computing costs decline, communications costs do not; and they form the base level for cost-per-terminal calculations.

MIX INSTRUCTION AND BUSINESS DATA PROCESSING AT YOUR PERIL

The mix usually fails; often expensively. Salespeople stress that such systems have succeeded but you should not depend on luck your first time out in the computer world. Instructional computing may depend for its success on how well each teacher uses your system from the other end of a phone line. However, a business data processing "system" includes prompt, reliable data entry and fast turnaround times. If your machine is supposed to print the district's payroll checks, it could mean your job if you fail; such conflicts may lead to instruction taking second place on your system.

WHICH SYSTEM SHOULD YOU GET?

Of course, you want to purchase the best available for your money. However, if you're just getting into instructional computing, you probably are not confident about what it is that you really need. In that case, you can save yourself some money and much grief by spending some time with people who have already been initiated. Specific actions that you might take include:

1) Invest a week in traveling to installations in your region and be sure to ask management, operators and users ---

    a) What are you trying to do; what is your mission?
    b) What would you do differently if you started today?
    c) What are your system's biggest strengths? weaknesses?
    d) How much does your system cost you to do a unit of your kind of work? (a very tough question)

2) Form a group of potential users to answer questions:

    a) What do we really want to do now? What later?
    b) What support will be required (funding, inservice, etc.)?

Note: The group will probably fail to reach permanent answers to these questions but it is essential that you share responsibility for answering them with the users.

3) Attend a conference, minicourse or institute dealing with the instructional uses of computing (See people in Appendix A; if you don't know anyone, you may wish to start with Tim Kelley or Dave Moursund or call the author of this guide).

4) Join the Oregon Council for Computer Education (O.C.C.E.) or the Association of Educational Data Systems (AEDS) and order a subscription to the People's Computer Company Newspaper

SELLING TIME: WOULD YOU RATHER BE THE LANDLORD?

If you sell time on your own computer, know that people will want more from your computer than you can provide. You will be accused of being partial to your own needs and empire-building with user fees. This will happen regardless of your performance.

Independent computing utilities must serve paying customers first and can avoid the political thicket. However, should you decide to become "the landlord", employ a full-time staff to serve users and do not ignore Larry Hunter's Law which states, "If you give something good away, it will be quickly used up and people will complain violently about the shortage." Thus, make fair charges for resources consumed (including staff and overhead) and don't stretch staff or computer time to cover too many users.

SOFTWARE

Most users will not see the pretty flashing lights of the computer console but all will have intimate contact with the software. Give it your closest attention when comparing systems.

DO-IT-YOURSELF? "BUT YOU CAN ALWAYS WRITE YOUR OWN PROGRAM TO DO THAT"

This remark is made by a salesperson who has a product that does not include some essential feature that your bid requires. The pitch depends on your ego to block out the shortcoming. Reputable vendors don't want naive customers to get in over their heads.

Good, reliable, well-documented computer software costs $10 per instruction in a competitive marketplace ...

```
10 PRINT 2+2
20 END
```

A $20 program, right? Not really. The catch is that programs that handle complex problems and/or interact with novices often require several hundred or several thousand instructions. Further, the art of writing reliable, not to mention efficient or aesthetically pleasing, programs isn't practiced widely or well and is harder than it looks to the tyro fresh from his first coding class.

Do not be misled by the ease with which a novice can code his first BASIC program --- Beginners All-purpose Symbolic Instruction Code (B-A-S-I-C). BASIC helps beginners rush into coding, but a lot of experienced programmers prefer other languages --- and not because they are snobs. Some complex ideas are harder to express in English than in German; some complex programs are harder to code in BASIC than in ALGOL, PL/1, APL, or COBOL.

A computer system includes hardware AND software --- insist on BOTH.

PROGRAMMERS ARE NICE FOLKS ... BUT THEY CANNOT TELL THE TRUTH

"But I can do it in five minutes ... no kidding!"

Don't you believe it! Books are starting to appear with titles like "The Psychology of Computer Programming." The optimism concerning the smallness of a task and the strength of their art is shared by nearly all programmers. Unfortunately, where reliable software is concerned, wishing does not make it so. But, be gentle in your scorn for the programmers' sins. Sometime in the next several years you are likely to catch yourself looking over your programmer's shoulder and saying, "Now that is really easy --- move over and let me do that myself."

"THE SOFTWARE FOR THIS MACHINE HAS NO BUGS"

This can only mean that the computer has no software or the salesperson has no integrity. If you think there is more to the product than its ads, talk to someone who purchased one a few years ago.

USER GROUPS: IS FREE SOFTWARE WORTH THE PRICE?

Probably "yes" if you are careful. The largest vendors have active user groups with offices supported by the company and/or modest fees. At best, user groups offer you and your students a source of free programs for your computer and a vehicle for passing around some of your best work. The personal contacts you will make with others facing problems similar to your own can be of more value than the free programs.

At worst, user groups can be a source of thousands of under-documented programs that "almost" meet your needs. Since developing good software can be more expensive than creating hardware, companies and users can benefit by the availability of programs which nobody has the responsibility to repair, but beware --- software breaks, and most good programmers would start from scratch rather than try to patch up an undocumented program.

WHAT BESIDES BASIC?

There are several computer languages more conducive to "structured programming" than BASIC. Writing time for languages is measured in work-years and so expenses have kept most educational vendors from worrying about anything but BASIC. However, some systems offer other languages. If you wish to offer non-BASIC languages, there are alternatives at one or more of the following levels:

1) Translators or interpreters written in BASIC --- at worst, these are unreliable "hacks" which offer only a superficial flavor of a language and devour huge amounts of computer time. Few offer more than an introduction, but some may be good instructional tools.

2) Limited single-user systems --- most general purpose minicomputers will offer a single user FORTRAN subset plus an assembler system.

3) A few vendors have minicomputers that permit up to sixteen users to run BASIC while a single user runs FORTRAN, ALGOL, etc., but full implementations of multiple advanced languages under time sharing on sub-$150,000 systems will probably not be delivered until late 1974 or early 1975.

WARNING: Careless or unscrupulous vendors have cultivated false hopes for true multi-language, multi-user systems on sub-$100,000 packages. The few inexpensive systems available are presently of more value in demonstrating different languages than in doing large amounts of computing. Don't plan on using such systems to accomplish the district's business data processing in FORTRAN while the computing classes use it for instruction with BASIC!

"BUT WE HAVE 'ADVANCED' BASIC"

All major vendors of instructional computing systems selling for more than $10,000 offer a version of the BASIC language containing enhancements to 1964 Dartmouth BASIC.

Regrettably, the originators of BASIC, Kemeny and Kurtz, were unable to impose standardization on succeeding versions and at this time every different manufacturer has at least one distinctive variant of Advanced BASIC which inhibits the exchange of program materials requiring advanced features such as alphabetic character strings, data files, and logical or matrix operators.

Fortunately, most government-developed curriculum materials use only the common skeleton BASIC, but watch out.

MANUALS COST MONEY (and some aren't very good)

You can buy independent texts to introduce languages but you must depend on your vendor to provide documentation for the operation of your particular system. Some vendors are noted for their expensive and poorly organized manuals --- get a full set when you first start shopping.

THROUGHPUT --- HOW MUCH WORK CAN IT DO?

Users of big computer systems evaluate computers as to how much calculation or data file manipulation they can do reliably per dollar.

They emphasize features that are not felt to be of great value by teachers. This section is not intended to talk you out of your values but to tell you the pros' point of view.

BATCH AND DEFERRED RUN

Jobs run under BATCH systems tend to run without human intervention until finished instead of wasting time swapping jobs in and out.

Interactive computing can cost you many times more than BATCH processing. You need interaction for computer-aided instruction but maybe interaction is not needed for teaching problem solving --- some people believe that interactive terminals discourage good programming practices in favor of "try something and see if it works" approaches.

Some computer utilities will let users request the "deferred run" of a job. For large computational tasks this can save much money --- the computer waits until a slack period, then lets a deferred job take over most of the computer as in BATCH. Sometimes deferred jobs are not done until evening brings a reduced time sharing load.

SCHEDULING ALGORITHMS --- RESPONSIVENESS VERSUS THROUGHPUT

A simple "round robin" scheduler gives each time sharing user a period of time such as one-tenth of a second in which to do the computing being requested and then turns its attention to the next user until, after at most several seconds, the first user is again serviced.

Such simple schedulers have the advantage of not permitting a large program to hog the computer, thus leaving the other users with an unresponsive terminal. This is probably an advantage in educational environments in which much of the value of the computer is its interaction with users. But, some such systems sacrifice doing "the greatest work for the greatest number" in order to remain always responsive. You should try to simulate your work load before deciding on just which you need.

SATURATION --- SHOULD YOU PLAN TO USE 100% OF YOUR COMPUTER'S POWER?

You should definitely not! Yes, it's a trick question. Users will be happiest when they do not experience the impact of big jobs. Time sharing depends on probability to scatter the big jobs through the day. Pity the poor computer system in which all users type RUN on their biggest job of the year at exactly the same time. You should hope that your system will almost always retain enough unused capacity to permit each user the illusion of being alone.

BUYING A COMPUTER SYSTEM

Appendix C lists a number of systems. This section deals with your negotiations with vendors' representatives. Get bids from several!

FIRST, GET THEIR ATTENTION

To retain much serious interest on the part of a good salesperson, you must be sure it is known that you are authorized to spend money. Salespeople advance according to successful sales, not prospects.

Remember that from your point of view, you are not just "spending money", you are "buying capability." This is not just a slogan!

It is best to define your needs rather than your budget limit --- few salespeople will quote a price on a system for less than your budget and most will assume you can be raised by twenty per cent.

When asked what your budget is, tell the salesperson that you will be taking bids and then be prepared to provide a detailed specification of your wants --- you may change them in later negotiations, but start by asking for what you really want.

BAIT-AND-SWITCH --- WAIT 'TIL YOU FIND OUT WHAT IT REALLY COSTS

Less honest than the "loss leader" pitch used in department store sales, the "bait-and-switch" technique brings you in the front door with an astoundingly low price that seems to apply to the package of the your dreams. However, "there are a few little extra costs." (!!!) The computer industry has many salesmen new to the business, so honest error is possible but is hard to distinguish from outright intentional deception. If you see a bad business practice, stop dealing with that salesperson and write company headquarters. You will be doing both yourself and the industry a favor.

"AND IF YOU BUY THIS MONTH I'LL THROW IN A GREEN WIDGET"

When urged to buy from a particular vendor because of extra goodies, be sure that you consider them valuable because you will probably pay for them. Also, get it in writing or they may never arrive.

BARGAINS --- "I CAN GET IT FOR YOU WHOLESALE."

Competition in the market place makes it unlikely that you will find more than a ten per cent cost variation in a piece of equipment that is new, a current model, with warranty, and immediately available. Big price cuts are available on discontinued or used equipment. Price and quality of used systems vary suprisingly.

Price is a function of marketability. Low cost may only mean that everyone wants the new model which has highly touted improvements about which you care little. High value per dollar can be had with such used systems, but the true bargains require some detective work, some care, and some familiarity with that model and its applicability to your needs.

Look for used systems from manufacturers' "used computer lots", from independents who specialize in used computer systems, or from individuals selling an old system that they have replaced. With manufacturers you take the least risk of getting a non-working system and with individuals you should pay the lowest price.

If you prefer a maintenance contract on your system, obtain written confirmation of acceptability from your maintenance organization before you write your purchase order and make the order contingent on prompt delivery of a working system --- try to get the seller to agree to pay all shipping costs if you have to return it because your maintenance people cannot install it satisfactorily within a certain time or within a set installation budget.

Watch out for bargains that have "trivial differences" from the standard model --- they may be part of a special order made for a now defunct customer and the trivial differences may keep you from running standard software.

WHO TO BUY FROM --- MANUFACTURER OR INDEPENDENT?

Larger manufacturers have full time sales representatives selling nothing but their own products. Experienced salespeople have often developed useful contacts in different parts of their organization and that can be of value to you. They should have good information on maintenance, software, and new products.

Independents purchase large numbers of systems to obtain quantity discounts of from twenty to forty per cent of list price. They can therefore quote you the same list price and make their money from their markup. Some independents are conscientious and will give you a lot of personal attention but there is a hazard --- they are often selling only hardware rather than a system. If you buy from them, you may have to pay thousands to buy the system software from the manufacturer. Ask experienced customers.

Beware of the special dangers of mixing vendors. When things go wrong (not "if they go wrong"!), vendors often point to the alien hardware or software as the culprit. This can cause the most catastrophic of delays --- months filled with headaches.

HOW CAN YOU PAY FOR A COMPUTING SYSTEM?

List price is what you'll hear first. Policies vary but many customers will be able to obtain at least a five per cent discount. Discounts go up if the vendor must win you over from a competitor, if the equipment is being discontinued, or if it is the end of the fiscal year (they want to make their annual report look good).

If you lease or get a bank loan, you will pay about 2.2% of the purchase price per month on a five year note. You may be eligible for a total payoff lease with escape privileges --- should the voters turn off the money. Otherwise, you may have a 10% payoff option at the end of the lease --- if you want to keep the machine, you pay one last payment (10%) and it's yours. Note, bank loans are cheaper.

A maintenance contract will cost you about 0.7 to 1.0% of the list price per month.

DON'T FORGET COMMUNICATIONS COSTS

Be sure you get a complete bid that includes all hookups and cables --- would you believe that some rather innocent looking cables cost over $50 per foot?

Multiplex devices permitting your computer to talk to many local or over-the-phone users come in many flavors. Some cheap "front ends" can handle a few terminals of the same speed without difficulty, but if you wish to handle different speed devices or more than eight lines you will want more exotic (and more costly) gear. Terminals may be rented from many sources --- look for a record of customers who are happy with their maintenance!

You can lose many thousands of dollars if you are not careful about the computer-to-telephone interface and the type of phone service you obtain. Get a bid from both the phone company and a couple terminal leasing companies before you decide on the interface. "Metered call rates" are lower than regular business lines --- the phone company tariffs (working rules approved by the Public Utilities Commission) permit such lower rates on phones that have fewer than 90 calls out per month. Your computer is unlikely to place calls, so you might save several hundred dollars per month on a big system. Often such rates must apply to an entire account; so be prepared to show that your computer consortium is a separate entity from your organization or combine all your lines --- 32 lines with 90 calls each might add enough to your switchboard pool to save your entire organization considerable money.

VENDOR/SYSTEM PERFORMANCE

"OF COURSE WE'LL DELIVER ON TIME, DON'T YOU TRUST US?"

Well ... no.

Indeed, nice guys get their computer last when they negotiate for a popular item. Even when dealing with the largest, most established firms, your purchase order should specify the latest acceptable date for delivery and user acceptance of the new system. Some vendors slip 60 to 90 days on most of their deliveries. Don't forget this point!

If you have the slightest intuition that they won't deliver, you should plan for the worst case --- try to give yourself 120 days padding so that you can replace a no-show with your second best bid.

"WHAT, INSTALLATION COSTS EXTRA?!"

You bet. Unless you have specified it, installation may be an extra charge and it could be steep. Proper handling of your order and signing up for a maintenance contract can prevent such extra fees.

VENDOR INTEGRITY

Some vendors have an industry-wide reputation for integrity --- some do not. Most pay salespeople on a commission or a partial commission basis but some do not. You are likely to receive less distorted information from those not paid on a commission basis.

USER ACCEPTANCE --- WRITE IT INTO YOUR PURCHASE ORDER

Since you will ultimately select a good computer system that meets your needs at the lowest cost, you can bet others have done the same. If you have violated Gunderson's Law, then you have probably purchased a product that has never been seen in the universe outside of an artist's mind. What might happen is that the salesperson gambled that, this time, the factory would not be nine months late delivering the new product or that you could be persuaded to wait "just another week or two" for delivery of your new computing beast.

THE WARRANTY

Be sure you get one and ask for a written description of exclusions. Some vendors' warranties are void if you use another vendor's equipment or supplies (tapes, disks) in their system. If you are caught sinning, you might have to pay the vendor hourly maintenance rates to fix your machine so that it will again be eligible for a maintenance contract.

"NEEDS NO OPERATOR, SUPERVISION OR MAINTENANCE"

Nonsense! If it moves, it breaks. Especially if it is a terminal, reader, punch, disk, tape drive, or anything else you depend on. It breaks sometimes even if it doesn't move.

To fix things that break will often cost you $35 per hour, plus transportation costs, unless you pay for a maintenance contract, which would typically cost 10 per cent of the new purchase price per year. Broken software will be fixed slowly, if at all, by any but the largest and most responsible vendors --- only a competent and permanent programming staff can provide such support.

Operation and supervision vary. Conservative systems, such as the Hewlett-Packard 2000 series, might run relatively unattended for time measured in months, from the moment they are uncrated. Sophisticated or flexible systems may need a full time operator.

DELIVERY METHOD --- Use Air Freight

Shop for the best route; but, unless you can save a lot of money, ship air freight for speed and good handling.

ALTERNATIVES TO BUYING A MINICOMPUTER/TERMINAL SYSTEM

Most secondary teachers entering the field of instructional computing in recent years have done so with a model 33 Teletype terminal (Teletype is a trademark of Teletype Corporation, Skokie, Illinois). The terminal is usually connected to a school owned minicomputer or a commercial service selling cheap time. This generally produces $3 to $5 per hour computing instead of the $6 to $15 per hour time sold on large commercial time shared computers. Even $3 to $5 per hour leads to about $4000 per classroom per year. However, there are alternatives to buying your own minicomputer and terminal:

PRESENTING COMPUTING CONCEPTS AND SIMPLE PROGRAMMING WITHOUT HARDWARE

Computers and terminals can and do motivate a great deal of student and teacher activity. Such activity sometimes gives little benefit beyond fun and group status. In the best situations, such fun and status give rise to educational programs that do use the computer to teach problem solving and to help motivate and assist other learning. However, the novelty will wear thin, and you will be able to see if there is enough depth of commitment to sustain a valid educational program.

If your budget does not permit giving you and your students direct access to computers, you may wish to present a "familiarization" class in which the students become acquainted with techniques for logically attacking problems and with some of the jargon used in the computing biz. "Computer Literacy" or "Computing Readiness" may be offered without machines.

If some cheap device is needed, several cardboard or plastic computers are available --- Bell Labs' CARDIAC has been popular with some students.

CALCULATOR VERSUS COMPUTER

"Calculator" no longer means "limited". If a calculator will meet your needs, it will do so for less money than a general purpose computer. Calculators usually have special purpose buttons and a fixed programming language, while computers usually use only regular typewriter keyboard characters. For numerical applications, multi-user programmable calculators will be cheaper than multi-user computers.

The smallest hand-held calculators sell for under $30 --- you might prefer to spend $4000 per year on calculators that can be used by all students at once rather than a single computer terminal; or perhaps a $4000 one-time-only expense for a powerful desk-top programmable calculator --- some of them even speak BASIC.

CARD READERS: EVERY STUDENT CAN RUN A PROGRAM EVERY DAY

When language coding is being taught, not more than five students can use one terminal to run and debug short programs. If you add extra terminals for punching up program tapes off-line, you may get a ratio of up to eight or ten to one with good typists.

Optical mark-sense card readers offer an alternative that can give a several fold increase to the number of programs run. Many teachers disdain cards because the computer then loses its immediacy or magic. Nonetheless, this approach is probably the cheapest way to provide real coding experience to a lot of people. Where motivation is very important (with non-science or non-math students), use terminals.

BUYING TIME: CENTRAL SUPPORT WITHOUT CAPITAL INVESTMENT OR BIG STAFF

A few commercial vendors sell cheap time. Careful use of the Oregon State University network will approach $3 to $4 per hour. Joining one of the other educational users' cooperatives mentioned in Appendix B-2 may also make such low cost time available to you.

Buying time from a computer utility has the advantage of giving you access to a central trained staff which can give classes to your staff and offer some limited programming advice, plus a shared central library of application programs.

Some commercial users will sell surplus time to educators at very low cost. Buying time rather than equipment can have the advantage of fixed cost and no risk plus no extra staff. However, administrators and boards of directors remain impressed by ownership and fail to realize that the computer box is only a token representing the "system" purchased. In addition, some people have observed that if the district signs a five year contract, it tends to give a five year commitment to the instructional program as well. However, most people do not use the flexibility available through owning their own machine.

APPENDIX A: SOME PEOPLE INVOLVED IN INSTRUCTIONAL COMPUTING

This appendix is omitted in this edition as the subject is covered in section XIII.E of this handbook.

APPENDIX B-1: Representatives of Some Equipment Vendors

This very limited list of vendors excludes those who do not have a manufacturer's representative in Oregon or have sold relatively few educational systems. IBM, and particularly Honeywell and Burroughs, might deserve more mention but they are not selling many low-priced smaller systems in the tricky educational market.

CALCULATORS:

Pocket or desk-top calculators are compared in the June 1973 Consumer Reports and the Texas Instruments Datamath TI2500 rated highest. It cost $100, but useful calculators were tested in the $60 range. Prices are dropping fast.

WANG LABORATORIES (Wang), 5319 S.W. Canyon Court, Portland, Or 97221 (297-2501) Frank Brandvold; Wang has a lot of experience using calculators in the classroom. They have a new BASIC speaking calculator which does not yet have much software, designed for use in the schools.

TEKTRONIX, INC (Tek), 8845 SW Center Ct., Tigard, Or 97223 (639-7691) Jon Gordon; Tek is not experienced in marketing to high schools, but their new line of programmable calculators is inexpensive and offers many advanced features. Tek products are noted for their exceptional reliability.

HEWLETT-PACKARD (HP), 17890 SW Lower Boones Ferry, Tualatin, Or 97062 (620-3350) Rick Baker; HP makes the world's most sophisticated pocket calculator and a wide range of desk-top calculator systems including the HP9830 BASIC language speaking calculator/computer. HP equipment is reliable and their sales and support staff is very familiar with the education market.

COMPUTERS:

HEWLETT-PACKARD (see above); Reliability, quick terminal response on small jobs when fully loaded with the normal mix of educational users, conservative development, and a large public BASIC library are the hallmarks of the HP-2000 series time shared BASIC systems. HP offers the single-user BASIC calculator mentioned above, plus 8, 16, and 32-user time shared BASIC systems. HP has sold a lot of HP-2000 systems and has not shown much interest in varying their offerings. HP's most painful language restriction is their failure to permit one or two dimensional string arrays. HP BASIC has only 6 digit precision and won't execute data files. Whatever its limitations, HP BASIC is probably used by more students than any other, and their salespeople are well tuned in to desires of secondary school teachers. The HP user group is well organized.

DATA GENERAL (DG), 5900 S.W. 150th, Beaverton, Or 97005 (646-5669) Bob Nelson; DG makes essentially only one computer, the NOVA, which was the first widely used sixteen bit machine. It comes in several versions from 4K and one terminal to a large 16-user system. Their computer systems tend to be a little cheaper than DEC or HP. Their Extended BASIC is available on some very small systems and is similar in power to HP BASIC. NOVA 840 systems are the first priced under $100,000 to offer a powerful 16 user BASIC with simultaneous batch that can handle a very nice ALGOL and an excellent FORTRAN IV. Performance on single CPU systems with over 16 users should be evaluated very carefully. DG claims they can expand in 16 user increments by adding a new CPU and swapping disk ($25,000 per increment); make them show you one in operation. DG does not have a well developed users group.

DIGITAL EQUIPMENT CORPORATION (DEC), 5319 S.W. Westgate Dr., Portland, Or 97221 (297-3761) Hank Merlitti; DEC offers a bewilderingly large assortment of educational packages from small to gigantic. Their systems are more flexible than HP's at most levels, because they are often purchased by business, engineering, scientific, and industrial customers in addition to their educational sales. At most levels, "more flexible" means that the operators often need to be more expert than on the same size HP system. A number of non-BASIC languages are available for DEC PDP-8 and PDP-11 minicomputers for single user mode; the same is true for HP and DG. DEC's 2-24 user time shared PDP-8 is the only multi-user multi-language system under $60,000 and boasts BASIC, FORTRAN, ALGOL, FOCAL and assembly language, although its BASIC does not compete in string and file handling with advanced BASICs of DG, HP, and DEC's PDP-11. DEC's PDP-11 RSTS BASIC-Plus is the most powerful mini-computer BASIC available, although its cheaper hardware configurations can be overloaded by a relatively small number of complex jobs. The Digital Equipment Corporation Users Society (DECUS) is well developed and well run; however, its educational group has not been its strongest section.

COMMUNICATIONS EQUIPMENT:

SIDEREAL CORPORATION, 1208 S.W. 13TH, Portland, Or 97205 (227-0111) Ray Zapp; Sidereal leases computer terminals and communications equipment. Call Sidereal, Western Union and your phone company business office for quotes on communications equipment.

USED COMPUTERS

AMERICAN USED COMPUTER CORPORATION, 15 School Street, Boston, Mass 02108 (617 227-8634) Sonny Monosson; AUC sells all sorts of used or discontinued stock. Sometimes it's a bargain and sometimes not. Send in your name to get on their mailing list. Make sure to get some sort of return privilege before you send in a purchase order.

APPENDIX B-2: TIME SHARING SERVICES

The following sell "cheap time" to educational users --- generally $5 per hour or less:

GENERAL ELECTRIC TIME SHARING, 2154 N.E. Broadway, Portland, Or 97232 (288-6915); GE offers a special service, BASIC-One, which permits unlimited use of one port offering a good BASIC and FORTRAN plus a large program library including a TUTOR package. One port means access to one of the phone lines connecting to the computer.

OREGON STATE UNIVERSITY, Computer Center, Corvallis, Or 97331 (754-2494) JoAnn Baughman; OSU has a very large computer and a gigantic on-line program library serving many parts of Oregon. Although theirs is a large CDC computer, OSU BASIC is very efficient and simple problem solving should not cost over $4-$5 per hour. OSU service has the advantage of expandability to many other languages and services, in addition to BASIC, plus their interest in teacher training. On the other side, OSU's system has the reputation of being loaded between noon and five, producing rather poor response times.

MULTNOMAH COUNTY INTERMEDIATE EDUCATION DISTRICT, P.O. Box 16647, Portland, Or 97216 (255-1841), Jack Slingerland; METCOM serves the Portland metropolitan area with very reliable HP-2000 C and F BASIC. METCOM has a large public library and specializes in serving its educational users --- educational users do not have to fight business or scientific data processing interests. The HP system retains good response time for small to moderate jobs, regardless of how heavily it is loaded. METCOM is supervised by Jack Slingerland, a dedicated and experienced Portland area math teacher.

O.T.I.S. (Oregon Total Information System), 354 E 40th, Eugene, Or 97405 (342-5361) Robert L. Dusenberry; OTIS offers HP-2000F BASIC as mentioned above for METCOM.

ROGUE VALLEY COMPUTING COOPERATIVE, c/o Ashland High School, Ashland, Or 97520 (482-4055) Keith Garrett; RVCC has an HP-2000E BASIC system, similar to METCOM and OTIS, but more limited in regard to program features and size.

PACIFIC UNIVERSITY, Computer Center, Forest Grove, Or 97116 (357-7037) Michael Clock; PU has a dual processor NOVA system that uses a very good BASIC of their own design. Michael Clock has a strong interest in teacher training and that should be of considerable value to educational users.

OMSI, 4015 S.W. Canyon Road, Portland, Or 97221 (224-9500) Rusty Whitney; OMSI has a PDP-11/45 which offers a very powerful BASIC language. OMSI presently offers low education rates but does not employ staff to support instructional users. Therefore, it can not be recommended to novice users of instructional computing.

APPENDIX C: A PRICE SPECTRUM OF INSTRUCTIONAL COMPUTING SYSTEMS

This incomplete list quotes rounded list prices which may be off by over 15%. 5 year leases cost $22/month/$1000 of purchase price; 3 year leases cost $32/month/$1000 (the bank is cheaper). Maintenance will cost 0.7% to 1.0% of new list price per month for a contract. Prices for multi-user systems are 'complete' except for phone equipment and terminals in addition to the one console terminal --- extra terminals cost $900 from Teletype Corporation or $1300-$1600 from your vendor.

GROUP 1: inexpensive Pocket or Desk-top Calculators

DATAMATH TI2500 ($70), Texas Instruments, Inc., Dallas; top rated small calculator in June 1973 "Consumer Reports"; add, subtract, multiply and divide plus constant storage; 8 digit readout

RAPIDMAN 800 ($50), Rapid Data Systems and Equipment, Ltd, Rexdale, Ontario; lacks a constant switch but is least expensive in the CR survey that included a decimal point

HP-45 ($325), Hewlett-Packard, Cupertino, Calif; the most powerful pocket calculator under $500; many math functions and several stored constants provided

GROUP 2: Programmable Desk-top Calculators

WANG 2200 ($4000 to $9000 and up), Wang Laboratories (see Appendix B for addresses for rest of systems); $9000 version includes 4K (4096) user memory locations, CRT (TV-like display), and IBM Selectric style typewriter output, keyboard and cassette tape for file storage; it speaks BASIC

HP-9830 ($6000 and up), Hewlett-Packard; $9000 version speaks a subset of popular HP-2000 BASIC, has a 32 character display and boasts a fast 240 line-per-minute thermal printer (takes special paper at 2 cents per page), plus a cassette for program storage; a plotter, card reader or typewriter output device may be added; matrix commands, strings, plotter routines, extended data file handling, and other functions may be added for $500 for each electronic module; user program memory can be expanded from initial 2K words at $1500 per 2K increment. HP can supply a program library and instructional materials for their BASIC.

TEK 31 ($3000 and up), Tektronix; $4500 version includes function keyboard, cassette tape for program and data storage, 4K of program/data memory, a fast and quiet 16 column alphanumeric thermal printer; extra devices include Tektronix 4010 graphics display output, X-Y plotter, more memory; does not program in BASIC and does not have a library of programs designed for high school use, but has its own algebraic language and a large library of math and statistics programs.

GROUP 3: Small Stand-alone Computers

EduSystem-5 ($6000), DEC, 4K general purpose digital computer with 12 bits in each word, plus a teletype terminal; paper tape programming system supports BASIC, a primitive FORTRAN, FOCAL (an interesting DEC "desk calculator" language that supports recursive subroutines) and assembly language; can be expanded in increments of several thousand dollars; for large expansion, the user should start with an Edu10, at $7000, expandable through the other DEC "EduSystems" up to the time-shared PDP-8 (Edu50).

NOVA 2/4 ($5000), DG, similar to above Edu5, except NOVA is a more sophisticated 16 bit computer; can be expanded to support a nice 4 user time shared BASIC system; if more expansion is envisioned, user should start with NOVA 2/10 at $6000

GROUP 4: Add Magnetic Tape or Cassette Tape Storage

HP-9830, see Group 2 for description; BASIC language capability qualifies it for Group 4 except that, unlike general purpose computers, it cannot be reprogrammed to speak other languages.

EduSystem-15 ($10,000), DEC; the next step up from Edu5 or Edu10 --- the addition of a single DECtape drive permits running a rather nice BASIC whose chief advantages include fast execution (it is a two-pass semi-compiler) and relatively large programs (6200 characters); disadvantages include lack of true strings and independent data files. Edu15 may be upgraded with an extra tape drive ($2500) permitting copying and "backups" or 4K more core ($2500) permitting use of Edu20 5-user BASIC (not "advanced"). Add both and get the superb OS/8 operating system --- it is only for single users but has many big machine languages and features.

GROUP 4A: BATCH --- Group 4 Plus a Mark Sense Card Reader

Add about $3000 to Group 4 for a reader and you can run "BATCH" --- the potential for feeding a stack of cards, containing programs encoded by the whole class, for rapid execution one after another, with results or error messages printed on the terminal. DG offers this feature starting with their larger Group 5 systems.

GROUP 5: Small Multi-user BASIC Systems

NOVA 2/4 ($9000) DG, 8K of memory and cassette for loading; supports a limited BASIC in fixed memory partitions permitting five users each about a 1K program (2000 character maximum); a $3000 upgrade to a 16K machine with two cassettes offers users a rather powerful "Extended BASIC". And the lowest priced swapping system permitting larger programs uses a 128K disk and totals $25,000 for the system.

EduSystem 20 ($8500) DEC, 8K of memory, system loaded from paper tape (inconvenient); up to five users permitted with 4K program partitions (8,000 characters); next fanciest multi-user system is Edu25 which starts at $22,000 with disk or magnetic DECtape storage of program and data files.

GROUP 6: Medium Sized Time Sharing Systems

NOVA 2/10 ($40,000) DG, a 32K 16-user system permitting Enhanced BASIC with good sized programs and powerful string and file commands. This system includes a one-half million word fast swapping disk plus two cassettes (extra cassette drives are $750 each). This is a very attractive system whose chief defect appears to be that it cannot be upgraded to the top-of-the-line NOVA 840 System; to get the same performance with upgrade capability will cost an additional $10,000 or more.

HP-2000E ($40,000) HP, although the 2000E limits users to 4K of program space (8000 characters maximum); the system is very "clean" and shows excellent response under almost all normal job loads even while supporting its maximum load of sixteen users. All programs written in HP-2000E BASIC will run on HP-2000F systems --- this is of vital significance if you plan to upgrade your system. Some of HP's competitors sell non-compatible BASICs.

HP-2000F ["Small F"] ($75,000) HP, supports sixteen users in good style using the very widely available HP-2000F BASIC language; a $35,000 upgrade will make this into the top-of-the-line "Big F" that handles 32 users; also permits addition of a lot of disk storage and even a line printer. 2000F BASIC permits 10K user program space (20,000 character maximum). HP has recently stated that the "Small F" will support 32 users in good style if those users are not doing a great deal of file work. See Group 7.

EduSystem-50 ["Time Shared 8"] ($40,000 and up) DEC; supports 4 to 24 users (more core should be added for more users at about $4000 per 8K of core); you should evaluate the Edu50 carefully for more than 12 users; TSS8, as it is known, is unique among small systems in that it provides each user a virtual 4K computer of his own with disk services and access to other peripherals such as DECtape and line printer if available; the user may use the fairly nice TSS8 BASIC or FORTRAN or ALGOL or assembly language. The rub is that a 4K PDP-8 is rather limiting and if a powerful string and file handling BASIC is essential for your work, TSS8 might not do it. With extra core and the several peripheral devices a computer science teacher might want, TSS8 systems usually come to about $50,000 to $60,000.

EduSystem-80 ["Mini-RSTS"] ($50,000) DEC; a PDP-11/40 based BASIC-only system for up to eight users. It features a very powerful version of BASIC --- the most powerful and flexible available on minicomputers --- but response time suffers if several big jobs are run alongside typical problem solving. $5,000 to $10,000 of additional memory will permit 16 users to run reasonably well --- the basic configuration includes 28K to 40K of memory and may be expanded up to 124K 16 bit words of memory. Additional peripherals (tape, card readers, printers, etc.) may be added. Installations seeking to expand to 32 users, including many compute bound jobs, should consider starting their RSTS system with a PDP-11/45 instead of an 11/40; this will cost about $7,000 extra.

GROUP 7: Largest Minicomputer Systems

NOVA 840 ($100,000) DG; a 64K system that permits a good 16-user Extended BASIC to run in one memory partition while real "batch" jobs in ALGOL or FORTRAN IV are run in the other partition. This system includes 10 million words of disk storage and a one-half million word swapping disk plus cassette tape drives, an intelligent multiplexer and a medium speed line printer; many other peripherals may be added (plotters, readers, punches, etc.). Vendors do not push the theoretical 32-user capacity of this system but suggest adding additional 32K CPUs and assorted support gear to the system for each additional 16 users at a cost of $25,000 per increment. Not many such over 16 user systems exist, so "buyer beware."

HP-2000F ($100,000) HP; a dual processor system that is the most popular "BASIC-box" in the country. The HP-3000C and now F have endeared themselves to educational users because they retain good response time even when fully loaded with 32 users! Response time is usually taken to be the delay experienced by a teletype user who has just hit the carriage return at the end of a line --- times in excess of two seconds start to become annoying, ten seconds is considered obnoxious and some systems do worse than that. Another reason for their popularity is that they are rather uncomplicated to run and have a much better "mean time between failures" record than their close competition. HP conservatism is the strength of the HP-2000F system and its weakness --- six digit precision is limiting and their failure to provide one or two dimensional string arrays on a system that is in such wide use is hard to forgive. Yet, its wide use is making it almost a quasi-standard educational BASIC language. Beware of running jobs requiring a great deal of random data file handling; when the 2000F is heavily loaded with other jobs, rapid processing of giant jobs will be sacrificed to maintain responsiveness and throughput on the "normal" jobs. The 2000F can add considerable disk space and can also make a line printer available to user jobs.

RSTS/E ($100,000 and up) DEC; if the system manager limits the degree to which a big job can steal the system, then a $140,000 32-user RSTS/E system built around a 72K PDP-11/45 having 16K of fast MOS memory is probably practical although the potential user should try out his job mix in a major test before buying. As stated elsewhere in this article, the RSTS BASIC-Plus language is the most powerful minicomputer implementation of BASIC now available. Unless limited by the system manager, user jobs may grab relatively large amounts of core plus peripheral devices in order to engage in pretty sophisticated data processing tasks in BASIC. Its file handling abilities and optional fifteen digit precision invite applications more complex than normal instructional or problem solving educational uses. By adding large 40 million character disk packs, line printers, card readers, and other commercial data processing peripherals, RSTS/E systems can quickly reach $200,000 and more for 32 user systems. RSTS's unusually flexible file handling permits reading and writing on 7-track or 9-track magnetic tape in most of the formats commonly used by IBM and other commercial data processing systems. Biggest gripe is that with 32 users, the OLD command takes forever on big files. Users are therefore urged to save COMPILED programs.

APPENDIX D: Some BASIC Language and Other System Features to Examine

When comparing "features" between different systems, this list might help draw your attention to points of difference between competitors.

1) Is machine binary, octal, hex?
2) What is its word length?
3) What operating systems available?
4) What non-BASIC languages?
5) Well organized users group?
6) How much can system grow?
7) How much is extra memory?
8) BASIC compiler or interpreter?
9) What peripherals available?
10) Does vendor discourage flexibility?
11) Mean time between failures?
12) Are data files compatible?
13) Are data files executable?
14) How many digits of precision?
15) Are numeric variables dimensioned?

16) How about string variables?
17) Maximum string length?
18) Midstringing (substringing)?
19) Program length limits?
20) CHAINing permitted?
21) Data file limits?
22) String functions?
23) String concatenation?
24) User defined functions?
25) CALL external subroutines?
26) Maximum number of data files?
27) Sequential files?
28) Record-oriented files?
29) Random access files (how?)?
30) Virtual arrays on disk?
31) PRINT USING (output formatting)?
32) Maximum number of users?
33) batch permitted?
34) Response time?
35) Hardware floating point?
36) Support high speed terminals?
37) List price of system?
38) Is software bundled?
39) Matrix operators?
40) Boolean operators?
41) Availability of good maintenance?
42) How much attention required by operator?
43) File security arrangements?
44) Cost per word for extra disk space?
45) Execution time for benchmarks typical of your job mix? (Be sure system is "typically loaded" during test)
46) How easy is it to crack user ID codes?
47) What sort of accounting/billing information is generated?

48) What is it that you really want to do that needs a computer?!

AFTERWORD

Thousands of teachers now purchase computing power to help them in the classroom --- most currently use the BASIC language on a minicomputer. The most important parts of the "system" include students, trained teachers, instructional goals, and the computing machines and the programs that tell the machines what to do. This Guide is intended to help novices in their encounters with vendors of computer hardware and the computer programs (software) most commonly used in schools.

Future editions will contain corrections plus additional sections. Early additions are expected under the headings:

Methods for Comparative Saturation Testing of Computing Systems
Results from Saturation Testing of Some Computing Systems
Computer Buzzwords

Please forward corrections and suggestions to the author in care of O.C.C.E. so that you can help benefit readers of future editions.

CREDITS

Vendors and O.C.C.E. members gave much help in preparing this Guide --- particularly Ashland High's Keith Garrett, U of O's Dave Moursund, and the Oregon Graduate Center's Roger Eiss. Cliff Moulton, Barry Smith, and Linda Craig of OMSI's Community Research Center also provided essential help as did OMSI's Virginia Kupfer and Dorothy Mason. Credit should also be given to OMSI for indulging the author's interest in computing long enough for him to learn what he knows the hard way and for encouraging him to write it down.

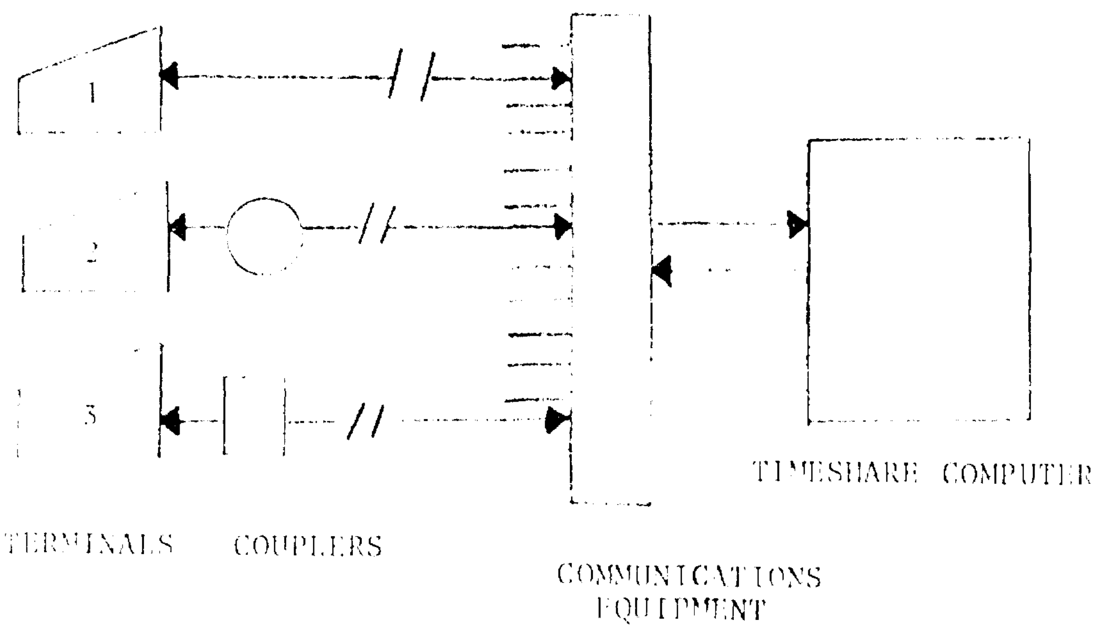
SECTION D: THE TERMINAL FACILITY

by Mike Dunlap

INTRODUCTION

This article discusses some of the common aspects of computer terminals. A computer terminal is a device that is used to communicate with a computer. It usually has both the ability to transfer information from the computer to the user (printer) and an ability to transfer information from the user to the computer (keyboard).

The diagram below illustrates the relationship between the computer and the terminal.



Each terminal is connected by a wire or set or wires to the computer's communication equipment. Messages are sent back and forth over the wire. Terminal #1 illustrates a terminal that is connected by a single wire directly to the computer. Terminals #2 and #3 illustrate the connection of a terminal through some type of switched network (phone lines). A small time-shared computer may support 2-32 terminals, while a large system might support 100-200 or more terminals.

This article is written for both the administrator and the teacher. It supplies information to the administrator who is considering the purchase, lease, or rental of a terminal. It suggests some operational considerations for the teacher working with a terminal.

This article describes the computer terminal and some of the considerations affecting its use. It explores three common terminals with a few of the available options. Then it discusses methods of connecting terminals to a computer. The housing of the terminal is considered next, including some notes on classroom location, laboratory location and general accessibility. For the person considering obtaining a terminal, there is a section on suppliers of terminals and who to go to for advice.

THE TERMINAL

In this section we are going to define three types of computer terminals and provide some insight into their nature and options. There are many types of computer terminals available, but we have limited this discussion to the teletype (TTY), cathode ray tube (CRT), and the cardreader. This limitation was imposed to keep the material relevant to computing below the college level.

A terminal is a strange animal. It comes in a large variety of sizes, shapes, speeds, colors, kinds, models, options, price ranges, and levels of intelligence. Terminals range from the very simple teletype (TTY) to the highly complex and intelligent machine which is in itself a computer system, but is used as a terminal.

It is a frightening experience, indeed, to walk through the forest of computer terminal options. When considering a terminal, one must decide: do you want a machine that is full duplex, half duplex, echoplex; a KSR or an ASR; does it need an acoustic coupler or a data access arrangement (DAA); tape or non tape; should the speed be 10 characters per second, 20 characters per second, 30 characters per second, or variable? The problem with computer terminal options is that they are as peculiar as any set of conditions you will ever find. Each computer system has a set of characteristics which make certain terminals better than others, and which make certain terminal options absolute necessities and other options completely unnecessary. This article will shed some light on a few of the available options.

The TTY

The teletype (TTY) is one of the cheapest and most available of terminals. It is also the device most commonly seen in educational envirorments. The teletype includes a complex printing mechanism that is not unlike an electric typewriter. The print head forms characters on a continuous roll of paper. The paper (hard copy) is then torn off by the user so he can carry his results away. The teletype has a keyboard that is nearly identical to the standard typewriter. As each key is depressed a code is formed and sent to the computer. The teletype must be placed upon a desk or table.

The teletype is probably the most available and easiest to repair unit used for computer communications. It will generally cost about $60-70 per month to rent a terminal, or about $1000 to purchase one from the teletype factory. In either case, if the terminal is not connected directly to the computer (connection kits cost about $100) then you will have to add to the cost of the device the cost of the telephone line. The line charges are often billed standard business rate and commonly run between $17 and $25 per month. Overall the cost of rental is usually estimated at $1000-1200 per year.

Generally one has several possible sources for acquiring a teletype. Many telephone companies rent or lease teletypes to their customers. Western Union rents teletypes at a very competitive price. They have special rates for educational institutions. In large metropolitan areas special service companies exist whose major activity is the lease and maintenance of terminal equipment. Within the metropolitan areas it is very easy to find service for teletypes. However, in rural areas it might be difficult, indeed, to find persons who could effect satisfactory repairs upon the teletype.

There are two main options that need to be discussed with the teletype. The first option is the teletype model. While teletypes come in many models suitable for computer communications, two of them, the model 33 and the model 35, provide the greatest number of features at the lowest cost. The model 33 teletype is a relatively small device.

The advantages to the model 33 are: relative portability, low cost and commonly available maintenance. The model 35 teletype is a much larger and more durable unit. It is three to four times the size of the model 33 and usually comes with its own stand. The model 35 teletype has far fewer maintenance problems than the model 33, but it is much more difficult to repair when it does fail. In addition, it is much more expensive to rent or to lease. Whenever you have a situation where good repair facilities exist and you need highly reliable terminals, the model 35 has advantages over the model 33. In those situations where the repair facilities are not so good, or where terminal failures are not critical, the model 33 is completely satisfactory.

Both the model 33 and the model 35 TTY's provide only upper case characters. A fairly recent addition to the TTY product line is the model 38. It provides both upper case and lower case characters, and is priced between the model 33 and model 35 prices.

A second option normally available on the teletype is the paper tape unit. The unit includes both a paper tape punch and a paper tape reader. Without the tape unit the teletype is a KSR (Keyboard Send and Receive). With the tape unit the teletype is called an ASR (Automatic Send and Receive). A tape unit model 33 is an ASR-33 and a model 35 without tape is a KSR-35. The KSR teletypes are significantly smaller than the ASR type machines. The tape unit has several advantages: it provides a way to prepare data when the computer is not operating; it provides auxiliary storage for programs; and it can be used to maximize the use/cost ratio of the computer, by punching programs and data off-line, and then calling up the computer only to transmit information at maximum typing speed. In almost all educational settings the tape unit is highly desirable.

In the use of the teletype one should keep in mind the considerations of portability, noise and durability. The teletype is generally not a very portable device. The model 33 and model 35 can be moved within a building if a strong cart is made available. The model 33 can even be moved outside of a building if one remembers that it must remain upright at all times (some things fall apart if it is not upright). A TTY is fairly heavy, so it usually requires two persons to move one any distance. In operation, noise is a problem. The model 35 is about 35% quieter than the model 33, but neither is quiet. In one study done of a small room containing 5 teletypes and no sound abatement, the sound level was found to be higher than the federal limitations for a diesel truck traveling at 60 miles per hour and thirty feet away. Generally, the teletype is a very reliable and durable unit, but like any mechanical device it absolutely must have routine maintenance. That maintenance is recommended every other month or every 750 hours of operation.

The CRT

Another commonly used terminal is the CRT. The CRT, Cathode Ray Tube, is simply a television set with an electric keyboard. Rather than printing by impact onto paper, the CRT forms letters by writing with an electron gun on the TV screen. (These terminals usually do not receive regular TV programs). With this method of printing, there is no available hard copy. In some applications this is a distinct advantage; in others it is a handicap.

The CRT is usually considerably smaller than the TTY. It is usually small enough to pick up and carry.

The cost of a CRT is extremely variable. It can be obtained as cheaply as a teletype for very restricted models and can range in price to over five-thousand dollars. The single greatest problem with the CRT cost involves the options. The "standard" CRT does not have a paper tape unit or provide hard copy. When these options are added they tend to inflate the cost of the device significantly. As with the teletype, if the unit is not wired directly to the computer, there will be line charges associated with its use.

Currently the CRT is not nearly as common as the TTY. In the future, however, the CRT will probably become the most common terminal in use. Because they are only used as computer terminals, the CRT is a little harder to find than the TTY. In most metropolitan areas availability is reasonably good and repair facilities exist. However, it is common that the terminal has to be returned to the factory for repair, although many of the companies are contracting with various local firms to do CRT service.

There are several common types of CRT terminals. The least expensive is the Hazeltine unit which costs about the same as a teletype and has rather limited capability. The limitations of the unit, however, do not interfer with many of the common educational applications. A second type of terminal is the smart terminal like the Datapoint 3600. It is an example of a CRT terminal that is actually a computer. A third type of CRT terminal is the graphics display terminal such as the Tektronix 4002. (These are manufactured in Beaverton, Oregon.) Within each class or type of CRT terminal, there are a multitude of brands, costs, and options. One of those configurations might be ideal for your application.

The outstanding feature of the CRT is the noise. Except for an occasional "beep" the terminal is completely silent. The silence makes it an ideal candidate for any situation where large volume of noise is unacceptable. It is ideal for use in the school.

The CRT is also a very durable machine. It is almost entirely solid state, and there are few moving parts. The only parts that actually move are the keys on the keyboard. With so few moving parts, the device is highly reliable and seldom needs repair. When it does fail, a circuit board can often be removed and replaced. The only routine maintenance needed is on the keyboard. Often it is not maintained at all but operated until it fails. This practice is dangerous with a teletype, but entirely acceptable with a CRT.

There are some very nice uses for the CRT terminal. First, the silent operation is a blessing. Second, the device usually is capable of changing speeds and can be adjusted to a higher rate of transmission, if your computer has multiple speed access. Third, the screen can be easily seen by others which makes it ideal for demonstration or for video taping. Fourth, the lack of hard copy has some distinct advantages: there is no paper to order, stock, and store; there are not stacks of used paper all about the room; it is perfect for testing when an exam needs to be kept secure; and it is ideal for programming and program modification.

The Card Reader

The optical card reader is a device that has significant potential in the school environment. The optical card reader obtains information by passing a card past some light sensors. Wherever a hole or pencil mark appears on the card, light is reflected to the sensors. The card reader converts card marks (standard card code) into the code used by the computer system.

The optical card reader has two main configurations. The first of these is the card to tape punch. Cards are fed into the reader and a paper tape containing the information on the cards is punched. This tape can then be fed into the teletype terminal for transmission to the computer. In the second option, the card reader works in conjunction with a terminal and is capable of transmitting directly from the cards to the computer. In this configuration, the terminal and the card reader are working together. If one does desire a paper tape from the cards, the teletype is placed in "local" mode and the card reader then sends the codes to the teletype paper tape punch.

The optical card reader is a relatively small device, about the size of a large bread box. It typically comes in desk top models.

The optical card reader reads standard card codes. This means that, usually, cards can be prepared on a standard keypunch, and that any pencil marked cards must be prepared in card code. The card reader then changes from these codes to the code usually used by the computer system (often ASCII). Pencil marked cards are easy to obtain. They usually contain marking guides and are printed with the appropriate codes for each letter. It is generally agreed that the cards are easy to mark and the skill can be easily taught to grade school children. Some people feel that the cards are tedious to mark, and that marking a large number of cards is a "pain in the neck." The truth of this contention does not obviate the use of such cards.

The cost of an optical card reader is in the neighborhood of $120 per month. Often it is desirable to lease or rent a unit but if a purchase is desired, the unit can be obtained for about $5500. Remember, no optical card reader will serve alone as a terminal. In every configuration it is necessary to have already available some other type of device such as a CRT or TTY.

There are two manufacturers of card readers that commonly work with schools. One is the Automata Company, which manufactures a card to tape punch. This unit operates at 20 characters per second, about twice the speed of a teletype. The other is the Hewlett-Packard Company. Its card reader interfaces with a terminal and can provide direct transmission to the computer. It is limited to 10 characters per second.

Card readers can often be repaired locally, if the manufacturer has a service outlet near by. In other situations, the reader must be returned to the factory for repair. The optical card reader is a reasonably durable device. It does require some routine maintenance, but not so frequently as a teletype. In remote locations, a teacher can easily learn to maintain and service it.

Because the card reader is either punching tape or working in conjunction with a terminal there is often considerable noise.

The optical card reader is usually able to read both 40 and 80 column cards. This dual ability is dependent upon timing marks which are printed onto the card. These marks tell the reader when to read information. In some situations, people have printed their own cards with much less than 40 columns. The cards can have a variety of special uses (i.e., test scoring, attendance reporting, sport statistics). The standard 40 column card is very easy for a student to mark, but if keypunched, requires that every other column be punched. The standard 80 column card is extremely difficult for a student to mark but ideal for the keypunch.

One of the fine features of the card reader is that the cards can be prepared by many people simultaneously in many locations. This permits the student to prepare programs and data away from the teletype, in study hall, at home, or even while on a date. It prevents delays at the terminal and makes the facilities available to a wider segment of the student population. With the card reader, the students in Business Machines coursescan provide the service of program and data preparation. Or, the football team statistician can keep statistics on cards during the game, and feed the data into the computer at a later time. The card reader is a device that extends the use of the computer throughout the school.

CONNECTING THE TERMINAL

This section will discuss the methods of connecting the terminal to the computer. It will consider hard wire and switched networks and the advantages of modems versus acousticouplers. The decision of whether the terminal will be connected directly by wire to the computer (hard wire), or connected through a switched network, will be made by the personnel in charge of the computer. If the terminal is to be hard wired, it will require a conversion kit so that voltages will be compatible with the computer input system. The advantage to the hard wired terminal is that transmission is reliable and the student does not have to contend with other people for access to the computer. When the terminal is hardwired it occupies one access path to the computer. This access path is only available to the single terminal to which it is connected. The hardwired terminal, then, has access whenever the computer is operational. The disadvantage to the hardwired terminal is that it is not portable.

A second method of connecting a terminal to the computer system is through the use of a switched network like the telephone line. In the switched network, a set of phone numbers are defined for the computer. The user obtains access to the computer by calling one of these numbers and then connecting the terminal to the line. If the phone numbers to the computer are busy, the user cannot gain access and must wait until a line is free. With the switched network it is possible to have terminals which are actually portable. The major disadvantage to the switched network is when all lines are busy and a user cannot gain access to the computer even though it is operational.

There are two common ways to connect the terminal to a switched network, the modem and the acoustic-coupler. The modem is an electronic unit that is physically connected between the terminal and the line. Modem stands for "modulator-demodulator" and it creates a frequency modulated signal which is sent over the line (in switched networks there is a modem on each end of the line). Even if there are a number of plugs to the phone line, the modem restricts the portability of a terminal. Usually it is connected directly to the phone line and thereby forces the terminal to remain in one place. The advantage to the modem is that it provides a highly reliable transmission of data. In some instances the telephone company will require that a device be placed between the modem and the line to isolate the phone line from any malfunction in the terminal. These devices are sometimes called data access arrangements (DAA).

The acoustic-coupler is a method of connecting the terminal to any single party phone line. The acoustic-coupler is a device that accomodates the telephone handset. The number of the computer is dialed and when the computer's carrier (1000 Hz. tone) is present, the handset is placed into the coupler. This coupling provides a physical isolation between the telephone line and the terminal thus oviating the need for the DAA. The acoustic-coupler performs the same function as the modem, and is actually a modem. However, it is the type of device that is extremely portable. It permits operation of the terminal wherever there is both a phone line and electric power. The problem with the acoustic-coupler is that of garbling and noise. The transmission of signals between the terminal and the computer are in the frequency range of human voices. If the room where the terminal is operated is extremely noisy, some of the background noise will be picked up by the acoustic-coupler. That extra noise will often garble the transmitted signal. The acoustic-coupler is also sensitive to bumps, knocks, and jolts. If one requires portability, however, the acoustic-coupler is the only reasonable method of connecting a terminal.

Related to the connection of a terminal to a line is the concept of duplex. In a half-duplex operation the keyboard of the terminal is connected to the terminal's print mechanism. Every signal that is transmitted to the computer is simultaneously printed on the printer. In this situation if garbling occurs on the line, the user is often unaware of it. Half-duplex signals travel in only one direction at a time. In the full-duplex mode the computer and the terminal are both able to transmit at the same time. In this operation, the user is aware of problems on the phone line because the keyboard of his terminal is connected to the computer and not the printer. Everything that the user types is sent to the computer and then transmitted back to the printer by the computer. Therefore, any character typed must make a complete circuit before it is printed. Garbling in the transmission is immediately obvious.

Some computers like the Hewlett-Packard 2000 series use a modified form of full-duplex called echoplex. In the echoplex operation there is no direct connection between the keyboard and the printer of the terminal. However, the transmission is only allowed in one direction at a time. The user still has the benefit of immediate feedback on a typed character, but he must wait until the computer is finished transmitting before he can begin a new transmission.

FACILITIES FOR THE TERMINAL

In this section we will describe the physical facilities necessary to house the terminal, the need for terminal supervision, and some of the materials necessary to operate a computer terminal.

The terminal requires both a location within the school and some sort of supervision. The location of the terminal is usually in a classroom or computer laboratory. The supervision is usually the responsibility of a teacher or teacher aide.

A considerable debate continues as to the amount of supervision needed for a terminal. It is obvious that a minimum supervision would include an occasional checking of the equipment, and appropriate replacement of paper, ribbons and paper tape. The maximum supervision would require a person present in the room with the terminal at all times. Some feel that the only way to operate a terminal is with maximum supervision. They seem to be concerned with theft, damage, and improper use. It is the author's opinion that the terminal should be operated with the minimum supervision. In five years of teaching computer in the high school, he is unaware of a single case of malicious damage. It is desirable, however, to have someone present near the terminal to assist students (students can do an excellent job of helping other students). In an environment of minimum supervision there seems to be a maximum of use. The major worries of theft, damage and improper use seldom occur.

Because of school laws related to general supervision of students, it is wise to have the terminal in a location where it can be seen by some adult. This casual supervision takes care of most of the "horse play" which would go on in the absence of an adult.

There are several reasons why a teacher's classroom is an inappropriate place for the computer terminal. First, the noise of the terminal itself or the commotion associated with it either effectively inhibit the normal work of the class, or the work of the class prevents access to the terminal at times convenient to the student. This placement effectively makes the terminal unavailable during major parts of the school day. A second problem is that of teacher dominance. When the terminal is placed in the classroom the teacher feels a responsibility to guard and protect it. Sometimes this protection actually prevents the free use of the machine by other teachers and students. A third objection to the terminal placement in the classroom is that of limited space. Most classrooms are too small to provide for a computer terminal and the necessary associated workspace. The terminal should be placed where students and teachers can gain access to it at all times without disturbing a class or class activity, and a place where adequate workspace exists.

The appropriate place for a computer terminal is in a laboratory of its own. The lab should be accesible to the hall, open before, after and during school, and should provide plenty of workspace with large tables. In this environment students and teachers are free to access the computer at their convenience. The laboratory placement encourages use by numbers of people throughout the school. In this environment the terminal belongs to everyone.

The computer laboratory needs some noise abatement facility. It is absolutely necessary to provide some method of decreasing the amount of noise associated with a teletype terminal. This can be accomplished by placing a rug on the floor, curtains on windows, and sound absorbing tiles on the walls. It is desirable that the room be quiet enough that the door to the hall can be left open at all times.

There should be several types of materials stored in a place easily accessable to the terminal. The materials include paper tape, paper, ribbons, pencil mark cards, and operation manuals for the computer system. A TTY terminal will use about eight ribbons, one case of paper tape, and three cases of paper in a year. These amounts vary with the application. Students are very capable of replacing paper, tape and ribbons. Make the materials available to the students and let them do this chore themselves as it is needed. Materials for the terminal are usually cheaper if purchased in quantity. There could be a considerable savings if one purchased three years worth of materials at one time.

OBTAINING A TERMINAL

This section makes some suggestions about obtaining a terminal. Here we will provide some suggestions on vendor selections, some vendors to consider, and some places to turn for additional information.

Once a need to have a computer terminal is fully established, the problem becomes one of selecting the best vendor. In selecting a vendor the person needs to have well in mind the type of terminal desired and the options that are necessary for his installation. Talk to several vendors and remember that the price is not the only consideration. Note that service, delivery date, and vendor reputation are as important as the actual cost of the machine. If you get a terminal at a cheap price, but you cannot have it maintained or repaired easily, what you have is an extremely expensive junk sculpture in the school. Remember that with the teletype routine maintenance is important be certain that you purchase a service contract on the terminal. (It is wise to get a service contract on any computer device in the school.) The service contract protects you against large outlays of money at unexpected times. Consider vendors who have good service contracts at reasonable prices.

Once a vendor is selected it is wise to check out his reputation. You may find that he is full of good promises, but never provides the services needed.

Watch out for multi-vendor problems. When the phone company supplies the line and another company the terminal, each will blame the other for any failure. It is more convenient to obtain teletype and line from the same company. However, it is not always financially feasible to do so. In such situations one is at the mercy of the service representatives of each of the vendors. The worst possible situation is to get the phone line from one company, the modem from a second and the terminal from a third!

There are several reliable vendors of computer terminals. One is the factory itself. The Teletype Corporation, 5555 West Touhy Avenue, Skokie, Ill. 60676, will supply you with a new ASR-33 terminal for about $1000 including shipping. However, it must be checked out by a reliable technician upon arrival, and it carries no maintenance contract. Note that the factory only sells terminals, it does not rent or lease them. A second reliable vendor is the telephone company whose business is the rental or lease of computing equipment. This type vendor usually has excellent mainterance contracts, and often supplies the best service.

When a person is considering the purchase, rental, or lease of a computer terminal there are many questions that cannot be answered in a short paper like this. In the event that you need further assistance and you live in or near Oregon, the following people will be able to help you.

Rusty Whitney
Director of Computer Activity
Oregon Museum of Science and Industry
Portland, Oregon

Jack Slingerland
Computer Instruction Specialist
Multnomah County I.E.D.
Box 16657
Portland, Oregon

SUMMARY

In this article we have considered the computer terminal. Several types of computer terminals have been described (TTY, CRT, and card reader). The connection between the terminal and the computer by hardwire or coupler was considered along with facilities for the placement and use of the machine. Some hints were given concerning the purchase, lease, or rental of a terminal, and a suggestion or two regarding the selection of an appropriate vendor.

The computer terminal has an important and impressive role to play in modern education. The appropriate use, supervision and location of the terminal will help to obtain maximum use from your computer system.

References

, "Hardware Educational Computer Buyers Guide", Peoples Volume One, Number One; October, 1972, pp. 13-15.

, "Teletypes Wi$e?", Peoples Computer Company, Volume One, Number Five: May, 1972, p. 23.

Watson, Richard W., "Remote-Terminal Communications", Time-sharing System Design Concepts, McGraw-Hill Book Company, New York, 1972: pp. 100-109.

Pacific Northwest Bell Telephone, Portland, Oregon.

General Telephone, Gresham, Oregon.

SECTION E: Operating A Small Computing Facility

by Bill LeMaster and David Moursund

Introduction

Computing facilities range in size from a single time-shared terminal to multi-million dollar installations. The medium sized computing centers such as OTIS (in Eugene), the Oregon State University Computing Center, and the University of Oregon Computing Center have budgets of approximately $1 million per year. Running such installations is a job for professionals.

This article is aimed at the amateur who is apt to be in charge of the computing facility located in a secondary school. The facility will generally consist of one or more keyboard terminals, or a minicomputer with one or more terminals, or one or more keypunches, and/or perhaps a mark sense card reader. This is what we mean by a "small computing facility."

The computing facility in a secondary school usually represents a significant amount of money, and affects quite a number of students. A single keyboard terminal connected to a small time-shared system such as one of the Hewlett-Packard 2000 series costs "someone" approximately $3,000-$4,000 per year. A single keypunch rents for about $75 per month. A mark sense card reader may rent for $125 per month. A lease-purchase contract on a one terminal minicomputer system may cost $250 per month. Any of these facilities could be used directly by, perhaps, a hundred or more students during a school term.

In most schools having computing facilities one person is usually "in charge" of the facility. Release time, or extra pay, for undertaking these responsibilities is rare, however. This article lists some of the activities that could be (should be?) carried out by the person in charge of the facility. It makes a strong case for providing release time for the computing facility director.

Scheduling Facility Use

A computing facility is a limited resource. Invariably at certain times demand for use of the facility will exceed its capability. What happens when the business class and the advanced math class both want to use the computing facility at the same time? The computing facility director should provide a plan for allocation of computing resources among the various users. He may need to monitor this situation on an hourly, daily, or weekly basis. He may need to work with the school administrator in charge of class scheduling to try to avoid serious conflicts in scheduling classes that will want to use computer facilities. He will need the authority to develop and implement an overall plan to be followed by all teachers and students.

Personnel

The computing facility director will usually have a considerable "staff" working for him. At his own level will be teachers of various courses in which the computer is used. Often these teachers will know how to run the computing facility and will supervise its operation while their students are using it. At a lower level will be student assistants. They will often be used to supervise other students who are using the facility, and to assist them when necessary. Viewed in this light a computing facility director may have a "staff" of a half dozen or more people "working" for him. This requires considerable organizational skills and supervisory time and effort.

Training and Consulting

The computing facility director is generally the computer specialist for the school, and serves as chief consultant to other teachers and to the school administration. He is expected to be knowledgeable in the pecularities and capabilities of the school's computer hardware and software. He is expected to be a resource person for the other teachers.

The computing facility director is usually also in charge of training of his "staff." Thus he trains student operators and student supervisors, and he trains his fellow teachers. As new facilities are added (software or hardware) and/or as new teachers come into the school it is the director's job to carry on the necessary training. Because of the relatively rapid turnover of student assistants this is a continuing task.

A very desirable situation is for the director to have time to contact other teachers on a one-to-one basis and to assist them in learning to use the computing facility for instructional purposes. This may involve giving demonstration classes and developing materials to fit a specific teacher's needs. Most schools grossly underuse their computing facility. That is, most schools that have computing facilities do not have enough teachers trained in its use and enough courses in which use of the computer is routine. In such cases there is clear need for in-service training as just discussed.

The Facility Library

A computing facility needs a library. It may consist of just an operator's handbook, and detailed information on the program library supplied by the computer manufacturer. Considerably more is desirable, however. There are several periodicals that are suitable for the secondary school computer library. There are many free publications of interest to students and teachers. There is a need to have reference books (books on computer programming and on various computer applications).

A key aspect of a facility library is a collection of well documented and readily available computer programs designed for use in the school. Some of these will be in the computer manufacturer's program library. Many will be student-written, or teacher-written, or written by the facility director. The facility director should continually seek good programs to add to this library. (Of course, as discussed in the previous section, he must continually work to acquaint other teachers of the availability and use of those programs.)

Managing the Facility

The facility director is in charge of making sure that appropriate supplies are available. These may include punch cards, paper tape, rolls of TTY paper, mark sense cards, TTY ribbons, etc.

A major aspect of managing a facility is maintenance and repair. Who does what when something goes wrong? This also includes such "minor" details as informing teachers who had planned to use the computer facility when it is unavailable because it is broken. The overall task of contacting a repair man and supervising his activity can be quite time-consuming. If the facility director can perform minor preventative maintenance functions and handle minor repairs it can save a school a lot of money, and make the computing facility much more available for use.

An important question to the facility director and school is "Does your school's or school district's insurance cover computer hardware and software?" Frequently this is not the case. What happens if vandals throw your computer terminal out the second story window, or pour acid into your minicomputer? Some insurance companies which do write policies to cover computing equipment are the St. Paul Fire and Marine Insurance Company of North America, and Royal Assurance of America.

Planning and Budgeting

The director is in the best position to know how well his facility meets the school's needs, and to plan for modification and/or expansion of the facility. The planning for expansion of a system should be similar to that used in acquiring a new facility. That is, it is quite time consuming, and requires good knowledge of the computer field. Even such a "minor" expansion as adding another terminal can be a major task. There are dozens of terminal manufacturers and perhaps hundreds of models to select from.

Budgeting may be one of the jobs of the facility director. This depends upon the particular school and school district. In any event the facility director will want to make a budget to reflect the true costs of the computing facility's operation. In particular this should include the salary for the fraction of the director's time which is spent directing the computing facility. It is interesting to include in this budget a realistic estimate of the costs of the "staff" (the free help one gets from student assistants, and from other teachers).

Conclusion

The main goal of this paper was to demonstrate that the facility director of a small computing facility has a large and responsible job. He should be appropriately compensated for this work. One way to do this is to give him significant release time--such as half the usual teaching load. An alternative approach is to give him extra pay, as is done for athletic coaches. The position of computer facility director is an important one, and the director should receive adequate recognition for performing this task.

BIBLIOGRAPHY

Books

Johnson, M. Clemens, *Educational Use of the Computer: An Introduction*, Rand McNally & Company, Chicago, 1971.

Post, Dudley L. ed, *The Use of Computers in Secondary School Mathematics*, ENTELEK, Newburyport, Mass., 1970. Chap 3.

Articles

Exelberg, Rodd S., (editorial), *Information and Records Management*, Vol. 5, No. 8, August 1971, p6.

Gilmore, Hal et al., "How Computers Are Changing Education", *School Management*, October 1970, pp 13-17.

Hoag, Ed., "Total Information for Oregon's Schools", *School Management*, February 1973, pp 37-39.

Moir, Ronald, "Planning For the Procurement of Pre-Programmed Packages", *AEDS Journal*, Vol. 5, No. 1, October 1971, pp. 21-33.

Pitstick, Les, "How Good a Data Processing Manager Are You?", *Computer Decisions*, September 1971, p 28.

Shields, Robert W., "What You Should Be Telling Your Computer People", *Government Data Systems*, May/June 1972, pp 26-28.

Spencer, Donald D., "The Computer Goes to School", *AEDS Journal*, Vol. 6, No. 1, Fall 1972, p 21+.

Landen, Dennis J., "Insurance For Your EDP Operation", *School Management*, April 1973, pp 26-28.

SECTION F:1 COMPONENTS OF A QUALITY COMPUTING SERVICE

by Fred Beisse and Richard Haller
University of Oregon

Almost every computer user has a definite opinion of the quality of the computing service that is being provided to him. This subjective opinion is often difficult to quantify, however. The purpose of this article is to list a number of components that might be used in measuring the quality of a computing facility.

Two major uses are envisioned for the material in this article. First, a computing center may wish to measure itself periodically to see if the quality of its service improves with time. Second, a school (which may already have some computing facilities) may be contemplating joining a computer network. The school, and other schools in the network will want to have a way of measuring the quality of the service they receive. They will want to make periodic assessments of user's opinions on how well they are being treated if they do join a network.

We have divided computing services into three major areas. The first is all those aspects which deal with the experiences of a user as he attempts to run a program. The second area deals with the necessary support for the ongoing operation of a Center or network. The last has to do with the attitudes of users toward their relationship with the network.

I. What is the nature of the computing facilities that are available?

   A. What facilities are available?

      1. Are the hardware facilities appropriate to solve the computational problems that need to be solved?

         a. How much main memory is available to computer users?

         b. What types of file storage are available (including magnetic disks, tapes, drums, etc.)? How much of each of these resources is available to users to store files?

c. What is the variety and accessibility of auxiliary peripherals (including 7-channel tape, plotters, storage scopes, etc.)? Are these peripherals adequate to the type of computing to be done?

2. Are the software facilities sufficient to meet the computational needs of users?

   a. Are supervisor or monitor programs useful to users and operators to describe the variety of computational problems easily? Do they permit sufficient flexibility to computer users?

   b. What languages are available for program development?

   c. Are good quality packaged systems available to meet user needs in all major applications areas?

   d. Do facilities exist which enable users to create their own program libraries?

   e. Can typical file management operations with utility programs be accomplished easily by users and operators?

3. What auxiliary computing equipment and services are available?

   a. Does the supply of well-maintained unit record equipment (keypunches, sorters, verifiers, reproducers, collators, etc.) meet user needs?

   b. Are data entry services (keypunching, scanner, character recognition equipment, etc.) and contract programming services available at reasonable costs to users?

B. How accessible to users are the computing facilities that are available?

1. Is the point of user contact with the facilities convenient to users?

   a. Are batch job submission procedures uncomplicated?

   b. Are timesharing terminal locations well-chosen to serve potential users?

   c. What time-of-day constraints are placed on users who wish to do processing?

   d. Are work areas and card storage areas available?

   e. Is the scheduling of user processing done with the requirements of all users in mind?

2. Is the software easy to learn and use (including both supervisor commands and applications programs)?
3. Is the hardware easy to operate?
    a. Are operating procedures for mainframe or RJE stations difficult for operators?
        1. Does operator training take protracted periods of time?
        2. Does the hardware facilitate jobs which require tapes and disk mounting?
        3. Do bottlenecks occur in the processing of jobs with card input and line printer or punched card output?
    b. Are timesharing terminals easy to use, especially for novice users?
        1. Is the number of terminals sufficient to meet user demand?
        2. Are the terminals well-maintained?
        3. Is the number of ports inadequate for peak use times?

C. How reliable are the computing facilities?
    1. What is the frequency and severity of hardware failures?
        a. What does the user need to do to recover from failures?
        b. Is preventative maintenance performed regularly?
    2. What problems are encountered by users and operators in the use of software?
        a. How mature is the software (especially operating systems)? Are crashes frequent and severe to users' work?
        b. What testing procedures were followed in the development of vendor and installation-supplied software?
        c. Are potential software problems well-documented for users?

D. How efficiently can work be accomplished from the user's point of view?
    1. Are timesharing and real-time facilities responsive to the users?
        a. What is the response time?

b. What is the throughput for individual users? For all users?

c. How bad do response times and throughput get under abnormal conditions and peak loads?

2. What is the turnaround time for batch jobs?

a. Does the scheduling of batch jobs recognize the needs of various classes of users?

b. How are very large, special, and emergency jobs handled?

c. What happens to turnaround under abnormal conditions?

E. What provisions are made for the security of user programs, data and resources?

1. What steps are taken to prevent accidents?

a. Are systems and public-access data files backed up? Are they stored "off-site"?

b. Does tape file protection include both operator and software checks?

c. What accesses of disk files are permitted and by whom? What steps are necessary to scratch a file?

2. Are steps taken to prevent deliberate violations of security?

a. What is done to prevent users from computing with another user's account?

b. Do administrative procedures protect the user's privacy?

F. What are the economic costs of doing computing?

1. How much does it cost to process jobs, store files, etc.?

2. Are accounting methods rational and understandable?

3. Are costs of auxiliary services within reason?

4. Are costs of converting processing from other computing facilities high because equipment and software are incompatible with industry standards?

II. Is support for the operation of the system adequate?

A. Is planning for future needs adequate?

1. Does the hardware configuration have adequate capability for expansion?

2. Does software development keep abreast of user needs and new applications?

3. Are the development staff keeping in touch with new hardware and software technologies?

B. How adequately are administerated responsibilities met?

   1. Are channels of communication between users and administration, such as a user advisory board, present and used?
   2. Is the administrative structure efficient?
   3. Is it easy for users to gain access to the right administrator when appropriate?
   4. Is staff morale high?
   5. Is the staff competent?

C. How adequate is documentation?

   1. Is it readily available?
   2. How accurate is it?
   3. How comprehensive?
   4. How recent?
   5. How readable?

D. How adequate is instruction in use of the facilities?

   1. Are the formal classes offered adequate?
   2. Are effective tutorials presented with reasonable frequency?
   3. Are useful self-help aids provided (includes computer assisted instruction)?

E. How effective is the consulting offered?

   1. Do users receive help in planning for future applications (e.g., grant proposals, data collection strategies, laboratory computers)?
   2. Are major substantive interests (i.e., social as well as physical science, business, administrative) reflected in the composition of the staff?
   3. How adequately are system problems handled?
   4. How accessible is the consulting service?
   5. How adequately are individual program problems handled?
   6. How frequently does the initial contact have to pass the problem on to someone "higher-up"?

III. How satisfied are users with their situation?

A. Are all users being treated equally and fairly?

   1. Are some groups at a given site given better treatment than others? (e.g., physicists versus business office)?
   2. Are some sites in the network treated better than others?

B. Do users participate meaningfully in decision making?

C. Do users feel that they can actually have an effect on the quality of their service?

D. Are users satisfied with the administration of the network?

1. Is the administration of the local site satisfactory?
2. Is the total network administration satisfactory?

E. Is useful accounting information provided with reasonable frequency?

F. Do users feel the charging system is fair?

G. Are potential users encouraged to use the system?

H. Are the interests of the community at large being served?

SECTION G: DISTRIBUTED AND NATIONAL NETWORKS

by Richard Haller and C.E. Klopfenstein
University of Oregon

Introduction

Today's electronics technology makes possible the design and construction of computer hardware whose massive computer power is contained in relatively small packages. A major portion of the cost of construction of these new computers is in the power sources, cabinets, and mathematical components, not in the logic components themselves.

Because of this, the computing power of a modern computing system increases faster than the cost of the necessary hardware. In addition the programs which control the operation of the computer (called operating systems, monitors, etc.) have become increasingly more sophisticated. As a result, attempts have been made to consolidate applications formerly done separately on several small machines onto one larger computing system. When this is done successfully, the total cost of computing turns out to be less than the sum of the costs of the formerly independent operations. The savings involved are commonly referred to as "economy of scale." The limiting factor for economy of scale is primarily the cost of developing more sophisticated operating systems (software) which make consolidation possible. In some cases this cost may exceed the savings involved or the system may be so complicated as to reduce the overall efficiency of the system to a point where economy of scale is not realized. However, in most cases, it can be shown to be more economical to combine as many computing applications as possible on a single computing resource.

The most sophisticated combination provides for a distribution of computer hardware components at various geographical locations, connected by high speed communications lines. Some sites which formerly had independent computing facilities now have input-output devices (called Remote

Job Entry or RJE stations) to computers located elsewhere. A network of computers that functions in this way can result in optimal usage of the computer hardware, since work loads can be distributed to locations where time and resources are most available. The redistribution of computing loads on the basis of demand is called "load-sharing". The physical location of each resource (node) is an unimportant factor to the success of operation of this kind of network.

To test the usefulness of the network philosophy, several experimental systems have been implemented at both state wide and national levels. Specific examples of each mode of implementation are given in the following sections.

Centralized Network

Central networks have been particularly favored in extending computing facilities to small institutions which have little or none of their own. A centralized network consists of a single, large computing facility in a single location with communications links to other sites (RJE stations).

Advantages: The advantages of this type of network organization lie chiefly in simplicity and economy of scale. 1) Since a single computer at a single site is involved, system maintenance and documentation is relatively simple. The user of a centralized system need only learn how to use one type of operating system and understand one set of documentation. 2) Economy of sclae takes advantage of the fact that the cost per "unit of computing work" becomes progressively less as the size of the system increases. By pooling demand and resources, participants are able to secure computing at a lower individual cost. 3) One large machine has the capacity to run a much larger program, if the machine resources are made available to it exclusively, than could be done on any one of several smaller computers with the same total throughput as the one larger computer. However, in a network with many different users and sites, it may be impractical to allow this except at unusual times of the day, such as 4:00 A.M. 4) Another potential of networking in general is dependent on the particular pattern of network use. Networks whose users demand peak at different times can more effectively utilize a common computing source. This is the case, for example, when the users are located in different time zones.

Disadvantages: Obvious disadvantages of the centralized approach include realiability and lack of diversity. 1) If all users are on a single system, then when it fails they are all affected. The use of duplicate components in a single system (dual CPU's etc.) or the linking together of several computers of the same type (a homogenous centralized net) are solutions to the reliability problem at additional cost.

Lack of diversity is the opposite side of the advantage of simplicity. 2) In designing computing systems, vendors are forced to compromise. The same design is not ideally suited to scientific uses, administrative or business uses, time-sharing versus large scale computation (numbercrunching) or printing. The type of system chosen by an installation reflects the relative interest level in different types of computing there--the specific applications required. This is particularly true in the case of higher education where different colleges and universities, while attempting to provide a variety of educational and research opportunities, are clearly much stronger in one area than in another. Systems of higher education such as that of Oregon, which to avoid duplication, concentrate certain programs such as the law school, medical school, agricultural prog[illegible] teacher education, etc., in different institutions, magnify this natural tendency to diversity of interest. A single centralized system, then, poses difficulties in providing adequate service to all types of users. The most difficult problems are undoubtedly human ones--providing a mechanism where conflicts of interest are worked out to everyone's satisfaction.

3) A not so obvious but very real disadvantage of centralized networks is their effect on user services at each remote site. Having systems, user services, and administration personnel all at the same site effectively provides for easy communication between groups. User problems encountered by one group become common knowledge, and their solutions are speedy. In addition, the level of knowledge of the user services personnel about the current state (i.e., problems and potentials) of the system is high. A user with a problem is more likely to have it solved by his local personnel. By centralizing systems, program development and administration at one site, very decisive and effective action must be taken to insure that local people have adequately informed local staff. Effective channels of communication have to be available in case the problem cannot be solved at a lower (local) level. Once again, people problems are the important ones.

4) In addition to the problem of education of local personnel, there are reasons to fear that the most able people will gravitate to the center "where the action is", further aggravating the problems of the remote user. This problem obviously affects potential users who have their own systems which would be lost by joining a network. Users without current facilities should, however, include them as a factor in evaluating the desirability of a local system versus joining a network. It is a part of the answer to the question "which gives me the most for my money?".

5) An additional problem involves the potential effect on computer science departments. Institutions which would lose computing facilities will also likely experience pressure on their computer science departments from two sources: a probable tendency for students, just as personnel, to want to be at the center rather than the periphery, and the loss of potential research assistantships to support students.

Existing Centralized Networks: Several networks, both commercial and educational have at one stage or another adopted the centralized format. Typically, the large commercial network format such as Tymshare, TSS, and CYBERNET, have moved to a distributed network format for reasons of reliability, communications costs, and potential load leveling. One area in which centralized networking is still popular is the Exploratory Program of Regional Computing Activities, conducted by NSF. Oregon State University is part of one of ten such projects funded. This approach has primarily been concerned with providing some sort of computing resources to schools which have little or none. While the results of the program were generally favorable, one interesting phenomenon was that as usage at the small schools increased, the most common response was to drop out of the network and get their own small machines. Apparently the advantages of access to a larger system were not sufficient to outweigh the desirability of locally controlled facilities. Since the cost of mini-computer systems has dropped dramatically while communications cost have not, in some situations it is not clear that networking is cheaper. This is particularly true when large distances are involved and large programs are not used.

Summary: A centralized system offers simplicity of implementation (lower costs), documentation, user education and administration. In some circumstances it may provide a larger system for a particular application on a special basis. Communication costs permitting, computing can be cheaper in a centralized network due to economy of scale than can several smaller independent operations. It raises a number of problems: communication with and education of remote users, satisfying a diversity of interests on a single system, and system reliability. It poses a threat to the continued existance of computer science departments at schools at remote areas.

Distributed Networks

In a distributed network, several computing facilities at different locations are linked together with high speed communication lines. RJE stations can access any of the facilities. Since one of the advantages of a distributed network is reliability and increased potential for load-sharing, this design is currently favored by national networks, both commercial and educational. Decreased communications costs over long distances and increased reliability of communications are also features of a distributed network which is attractive to national users. There are two kinds of distributed network: homogenous, where all the computers providing service are of the same type, and heterogenous, where they are not. Both kinds of distributed networks offer the advantages of reliability and potential load-sharing. They allow a degree of local control and hence the possibility of some degree of specialization at each site in response to local needs. The presence of system, program development and administrative personnel at each site helps maintain a higher level of user services as the price of some increase in personnel costs. Existing computer science programs, far from being threatened with the loss of support for students and the exodus of quality students and staff, would be strengthened by the presence of a distributed network since a wider variety of resources are avaliable to them. This is particularly true in the case of a heterogenous network where users would be able to gain experience on different machines and communication with counterparts at other sites would be stimulated.

Advantages: 1) Homogenous networks offer the advantage of simplicity of implementation, documentation, education and charging structure. Programs or data sets (files) created at one center are immediately usable at another. One set of documentation is sufficient. Designing a protocol for communication from one center to another is straightforward. As in a centralized system, the user needs to know only one set of procedures for using the resources of the system. 2) Since more than one computer is available to serve users, failure of one does not mean that all users are affected. A distributed network, in fact, provides more reliability than independent computers, since users at the site which is not working have alternate sources of computing available. Computing continues to go on, though perhaps at a slower rate, at all sites. The ability to respond to the failure of a component by reduced services rather than their complete lack is sometimes called "graceful degredation." 3) Heterogenous distributed networks offer the ultimmte in flexibility of facilities. This advantage, though it springs from differences in emphasis at different centers, is nonetheless of value to all users of the network. For example, physicists at one university or research center may not have enough influence or resources to provide the kind of computing they would like at their own center. They may be able to make good use of those created at another node in the network by a larger, richer department. 4) This advantage extends to the instructional level as well. Special programs developed by one department for instructional purposes are available to others. The main point is that decentralization and a degree of local control encourage diversity while the presence of a network helps insure that these diverse applications are usable by others. Lack of transportability of software developed on one machine to a different one has led in the past to a situation often described as "re-inventing the wheel."

5) An additional advantage of heterogenity lies in the potential for expansion. Once the problem of getting different computers to talk to each other is solved by specifying standard protocols and interface hardware, it is relatively easy to add additional computers to the network, regardless of their design. This relieves the network of dependence on a single manufacturer and allows it to take advantage of newer, faster and cheaper hardware when desirable. This advantage is an important one since the trend in the industry has been to more powerful and less expensive machines. Commitment to a centralized network or a homogenous distributed network means that to take advantage of newer, cheaper technology, the entire network may have to be replaced. 6) The final advantage of a distributed network, especially a heterogenous one, is the protection it gives to the users. A centralized net is like a monopoly utility. You either use the services provided or "take your business to Walgreen's." To deal with this problem, regulatory bodies are set up, supposedly representing the interests of all customers. Another solution to this problem is competition and the forces of the market. If suppliers must compete for customers, they are more likely to actually listen to them. Both types of organization have been both successful (Bell Telephone, the airline industry) and unsuccessful (U.S. Post Office, the auto industry). The main problem in a market place solution is to insure that the suppliers truly compete with each other. Given the choice, most users, we believe, prefer competition, while established suppliers perfer monopoly! In addition multiple suppliers means that each does not have to be all things to all people. Needless to say, the larger the number of suppliers, the better this solution works.

Disadvantages: 1) The main disadvantages of a heterogenous network lie in the difficulty (and thereby the cost) of their initial implementation and the education and 2) documentation difficulties posed by different systems with their own pecularities. Although implementation of a heterogenous network will always be more complicated than a homogenous one or a centralized one, there is reason to believe the gap will soon be closed considerably. The very nature of the problem forces solutions in the direction of standardization of communications. Once large, diverse networks, such as ARPA (to be discussed later) are successfully implemented, their hardware, protocol, specifications and in many cases, even programs, can be utilized to construct smaller networks with a different mix of computers.

Education and documentation are definite problems. In existing networks of this type, in order to use a particular system, the user must be acquainted with all its vagaries. There is no standard operating system language. Even languages with the same name such as BASIC, ALGOL, FORTRAN, or COBOL differ in their special features on each machine. That is, of course, one of the prices of diversity and is in itself an educational tool since students cannot predict what system they will be working with after graduation. In that sense it may be a blessing in disguise. Typically on such systems the user masters one which he believes is best for him, probably the local one, hears about some nice feature somewhere else, and becomes motivated to learn how to access it. The primary task of good network administration in this environment, then, involves the "advertising" of the resources available at all the different sites and seeing that appropriate information for accessing and using them is readily available.

3) If the network is homogenous, flexibility, though greater than in a centralized network, since each site does not have to be identified, is s ather limited.

Examples: The best known homogenous distributed networks are probably IBM's, TSS and CDC's, CYBERNET, which are used primarily for business applications. A useful example in education is the MERIT (Michigan Educational Research Information Triad) network. It consists of an IBM 360/70 at the University of Michigan, another at Wayne State University and a CDC 6500 at Michigan State University. It's organization is quite similar to the nationwide ARPA network which is discussed separately under national networks. Each computer is linked to the other via a PDP-11/20 communications controller, mini-computer via dial up voice grade telpak lines. The primary problems encountered in this project have been management ones. In particular the issue of cash flow from one university to another is a knotty one. These problems have not prevented MERIT from being a success as far as its participants are concerned.

Summary: In summary, distributed networks offer the advantage of reliability and potential load-sharing. They diffuse expertise more efficiently to all users at the price of higher personnel costs, and provide more opportunity for local accomodation to local problems. In particular heterogenous nets promote a diversity of applications and enable a high degree of future expandability. If long distances are involved, substantial savings in communication costs are possible. The main disadvantage of heterogenous networks is difficulty (and cost) of implementation, documentation and education. Implementation costs should decrease as more heterogenous networks are formed.

National Networks

Most existing national networks are commercially operated and are primarily business oriented. While the services of such networks might be useful for administrative functions, their use for instruction and research applications is questionable. In the absence of significant educational discounts it is also questionable whether commercial networks would be economically feasible.

An exception is the ARPA (Advanced Research Projects Administration) Network. This is a distributed, heterogenous network linking up some of the most sophisticated computers in the nation, including the massive ILLIAC IV. At least 15 sites and 23 computers are currently part of the network. Although origina-ly a research project in the feasibility of such networks, sponsored by the Department of Defense, plans are underway to convert it to a self-maintaining national network. Interesting features include: the incredible power and diversity of the machines available, the highest reliability of any communications network in existance (one bit error in 10^{12} bits transmitted and alternate paths in case one connection is broken), low cost of data transmission and high data transmission rates. For the more sophisticated user, the capability of coordinated action of several computers on the same problem is being implemented.

Cost of tying into the ARPA Network is $50,000 for an interface message processor (IMP) which can be interfaced to up to 4 local computers at a cost of $15,000 apiece. In addition, maintenance would run about $5,000 a year, and communication costs are $16,500 a year plus 30¢ per million bits (kilopacket) beyond 4500 kilopackets in a single month. Additional costs might be incurred in software development for the local computer, but due to the large number of different computers currently in that net (including PDP-10's, IBM 360/70's, PDP-11's) it is highly likely that the necessary software could be obtained at little or no cost from a current user (since the development was government sponsored).

Other national educational networks in the planning or development stage include the Quantumn Chemistry network and the NSF national network.

The former is, of course, restricted in its general usefulness.

Advantages and Disadvantages: 1) The disadvantages of national networks are the same as those of any distributed, heterogenous net, including complexity of documentation and education. 2) In addition there is the problem of cash flow to out-of-state institutions and its possible bad effect on local computer centers. 3) On the other hand the large number of centers in such a network competing to give service helps insure reasonable charges for computing and motivates suppliers of services to be friendly and attentive to the needs of their customers, unlike the monoply, "take it or leave it" situation inherent in centralized nets. 4) Perhaps the greatest advantage is that users can tie into the net at a cost in the case of ARPA which is realistically affordable by even a small school and have the use of resources affordable only by the largest and richest schools. 5) The effect on computer science departments ought to be similarly benificial. Although students will still tend to gravitate to the schools where the large computers are actually located, no small or even medium-sized operation could hope to compete on equal terms with these giants in any case.

Section H: Abstracts of related articles for Chapter VIII

Unsigned article
Computer Co-op: Share the wealth and Cut the Costs
School Management. Vol. 16, No. 10 (October 1972), pp. 20-22

Both the public and school administrators find it difficult to accept the price range of exclusively owned computer equipment suitable for schools today. Prices appear to range from $75,000 to $2,000,000. Schools and colleges in northern New Jersey have formed a co-op to cut these costs significantly. Thirty-five schools and colleges comprise the Instructional Computer Cooperative, IND, (ICC) built around two Hewlett-Packard time shared systems.

Thirty-two terminals can be used simultaneously, connected by normal phone lines. All operating costs are shared on a flat fee basis. When ICC, an independent, non-profit group, was formed, sixteen schools participated on a 24-hour-a-day, seven-day-a-week connection for a fee of $3000 per school. Today twenty-four schools belong and the fee has been reduced to $2000. The cost is low because 1) the systems themselves are relatively inexpensive, and 2) ICC's staff is composed of volunteer teachers. Also, the Hewlett-Packard equipment functions unattended.

ICC was formed after many member schools found the cost of time-shared greatly exceeding the expected budgeted allotment of funds, as a result of an unexpected high usage rate. The first thirty-two terminal system was purchased with a $35,000 NDEA grant. The second was purchased to allow for increased membership and to decrease long distance phone bills. At the present time, a centrally located multiplexor funnels long distance computer messages from several users over one telephone line. This will be eliminated next year when a third system will be incorporated into use. Based on past experience, this new system will be self supporting after one year.

ICC hopes within the near future to offer services to other scattered, lightly populated areas of New Jersey. The toughest problem is convincing other districts that a co-op organized and run by school people can function efficiently. Several services are offered to new members, including instruction in using equipment, workshops, and an "Instructional Dialogue Facility" (IDF) that enables subscribers with no knowledge of programming to write their own lessons.

A series of new uses for the computer in education is being studied by ICC and a list of several of these new uses is included in this article.

Stewart Weimer

Butt, R.E. & C.L. Guthrie
Can Small School Systems Use Computers?
School Management (August 1972) pp 37-39

A new dimension in management and educational concepts has been developed in Loudoun County, Virginia, a metropolitan area of Washington, D.C. The project serves as an excellent example of what can be done for smaller school systems.

Detailed analysis of all aspects of computer uses and goals were set up within definite parameters. Goal setting and problem definitions were developed for the total system. The objectives were to obtain a responsive service to the educational system and community and yet remain within the severe cost limitations.

An important result of the study was the combining of educational and management concepts in order to develop performance requirements for the computer. Another result of this concept was a decision to use a "teacher-manager" to provide an effective bridge between professional teaching objectives and management objectives. To help decrease costs, the decision was made to not use the computer for smaller or seldom-occuring problems. Rather, programs were developed to handle the 85%-90% of standardly occuring problems.

One purpose served by the Loudoun Project has been to provide a model which can help other small school systems avoid most start-up costs and to avoid the "large computer syndrome", the need for a bigger and better computer.

Conclusions reached by the Loudoun Project and others are: most schools and county systems can afford and utilize a small general purpose computer following the Loudoun model; the critical area is the application of management techniques rather than development of computer technology; priorities and needs of small schools and governments tend to be more in the areas of fundamental workloads and community relations; sharing is entirely feasible; there are ample condidates for sharing, such as hospitals and libraries; general purpose computers are within reach of small communities; human factors are more significant to the success of a small computer program than technical factors; prohibitive start-up costs are reduced; the key roles in bringing computer support to most counties and school systems must be taken by members of the governing boards and a few top officials.

William LeMaster

Gilmore, Hal, et al
How Computers Are Changing Education
School Management, (October 1970), pp 13-17

According to a CAVI survey conducted on computers in education, the greatest use is for administrative purposes, followed by classroom instructional use, with combined instructional and administrative use at the bottom. The survey also showed that computers tend to concentrate in larger school districts. Almost half the total number of computers used by schools are concentrated in about 17 per cent of the school districts. The most common methods used to gain access to computers are renting and time-sharing, with about 64 per cent leasing or renting and a small minority subscribing to time sharing plans.

The authors of this article discuss case histories to show a variety of ways in which a school can pay for computer capability and ways in which a computer can be used.

How to start a computer cooperative is illustrated in the development of the computer center for the Intermediate School District 109, located in Everett, Washington. The center provides support service to 16 local school districts spread across Snohomish and Island Counties on Puget Sound.

Elementary School District 161, Flossmoor, Illinois, used the computer to project enrollment changes in their school district. The authors discuss the planning involved and the results of their survey. The techniques employed in conducting the survey would be appropriate for many school districts.

San Ramon High School in Danville, California, started a computer education course for $6,000. The author discusses the development and operation of the facility.

In Danbury, Connecticut, the school district and the city government combined resources to streamline their operations with a computer, an item which neither could afford alone. The cost of the company operation is evenly divided between the district and the city, although school applications take up about 70 per cent of the computer time.

Bill LeMaster

Gross, Andrew C.
Accessibility and the Small Computer
Datamation Vol. 17, No. 22, November 15, 1971
Pages 42-48

The 1970's will be a decade of growth in computer accessibility. To be effective, access must be available at normal work stations and offices. This will improve efficiency and permit true man-machine interface to develop. The demand for access will account for growth of both small computers and terminals. This will be the decade of the user.

Large time-share systems have significant advantages over small computers in terms of large data banks, computing capacity, and variety of peripheral equipment. Costs and waiting time for users continue to be reduced. Free standing small computers, on the other hand, have advantages in rapid access time, avoidance of red tape, and protection for confidential data files. There is prestige in owning a computer and being able to work with hands-on. In full time use costs can be less than time-share systems, particularly in locations where there are high communication costs. In mobile applications, small computers have no competition.

The advantages of both types can be obtained by using a mini-computer as a terminal to a large computer.

Growth in shipments of computer system hardware will triple from 1970-1980. Small and mini-computer units will comprise about 10% of sales throughout the decade. Although the decade started with about 100 producers of small and mini-computers, mergers will reduce this number to less than 50 in the U.S. by 1975. The number of unit shipments of U.S. small computers will increase about 7 times during this decade, but the dollar amount will only triple. This is due to declining prices and a shift in emphasis to smaller units. Small and mini-computers will experience good growth abroad, but U.S. firms will encounter stiff competition from foreign manufacturers. Scientific and engineering uses have dominated the computer scene in the past, but there will be a shift by 1980 so that the data handling market will account for over half of all applications.

Robert Christensen

Heilman, Robert L. and Moursund, David
<u>So You're Thinking About Getting A Computer</u>

In years to come we can expect that many educational institutions will be in the process of acquiring a computer. A major decision that will be needed to be made is whether the proposed system will be for both administrative and instructional use or for just one of these.

The computer acquisition process begins with a needs assessment. For many schools it may end here! There is a good chance that switching to using a computer will cost much more than is now spent without a computer. Each potential use of the computer should be examined and this use projected for up to five or more years into the future. The needs assessment should be carried out by knowledgeable staff who are giving time to interact freelv with potential users of the computer system. It is sometimes/often desirable to hire an outside consultant in the needs assessment.

The next step is to specify what the computer should be able to do. Choice of proper languages, number and types of peripheral devices to be used, type of process (batch or time sharing), and other decisions must be made to provide information to computer vendors so they can make accurate bids on the necessary facilities. These decisions should be general enough so that one's specification guarantees as much competition between vendors as possible, in order to keep costs down.

Along with the bid a vendor should provide detailed technical information, including manuals on the facility being offered. He should generally provide access to a similar system so that one could visit and use the system offered. In dealing with a vendor one may need to reassess one's needs and specifications so as not to purchase extra features that are not worth the additional cost.

Perhaps the single most important idea is that the computer acquisition cycle should be under your control. You (not your "friendly" computer salesman) should direct a careful needs assessment. The needs should be translated into a general statement of computer facilities needs. The computer vendors should propose solutions to these facilities needs. You should determine the best (for your purposes) of these proposed solutions. The final purchase contract should be designed to protect you as well as the computer vendor.

James E. Muck

Kenny, J.B. & R.R. Rentz
Planning for an EDP System
School Management, (October 1970) pp18+

A common misconception held by educators is that in electronic data processing one should "get the hardware" first, then identify information needs and simply automate existing records. A better procedure to prepare for EDP is suggested below in this article.

The superintendent should support a study of the information needs of the system. His help is essential since he will be the person who decides on the priority for items to be automated.

The principal then identifies information needs within the school. Listings should be obtained from the professional staff and the clerical staff. He should correlate these with his data to provide definite classifications of information. This will also help him to identify different classes of users.

A person or management consultant firm should be retained by the superintendent to perform a systems analysis of the data needs of the school system. This group should work closely with the principal, cheif guidance person and staff. In this way, staff will feel they had a voice in the planning and both the systems analysis group and staff can identify data that are critical to the work of the staff. A school system of 15,000 children can expect to pay between $6,000 and $30,000 for a feasible study, depending upon the depth of the study.

Once the data needs have been identified, the superintendent should bring in a third group of unbiased people from a local college or university to attend meetings of the central office staff, the systems analysis group, key personnel from the separate schools and hardware manufacturers.

Based on the study by the systems analysis group, certain hardware configurations will be recommended for lease or purchase. The authors recommend only small or medium size computers since the amount of data will continually increase. Once the general size of the computer has been determined, the third party should be called in to help. Hardware manufacturers may try to oversell the needs of the school system and the help of this group can be very valuable.

After requests have been let to the hardware dealer, a competent director or supervisor should be retained. He should be on the job 6 months prior to delivery to get things organized and to "dry run" the system prior to installation of the machinery. As soon as the system becomes operational, lines of communication between the various user groups and the data center must be developed.

Bill LeMaster

Meyrr, Donald P.
Yes, We Have No Computer
Community and Junior College Journal, Vol. 43, No. 9,(June/July 1973)
Pages 18-19

When a college begins a data processing program it has been traditional to lease a small computer, then to begin almost immediately to order a larger, faster, more flexible computer, and finally to hire a data processing staff which would proceed to re-invent the wheel by developing their own data processing software systems from scratch.

Choosing to follow a non-traditional method, Johnson County Community College, located in the Greater Kansas City area, today leases enough hardware to carry out the instructional function for over 3,000 credit students and over 2,000 community adults enrolled in continuing education. At the present time, the college leases the software and computer time using a sliding scale with the cost per student decreasing as enrollment increases. The total cost is slightly more than $20,000 annually.

The college also leases an online (telephone) student personnel administrative software package and computer service. The package permits data to be entered directly into the computer from a keyboard attached to a Cathode Ray Tube. After the input data is placed on the CRT screen it takes only three to five seconds from the moment the send button is depressed until all the data from the screen are transmitted to the computer and the files are updated. If data are incorrectly transmitted, an error message will appear on the CRT screen and the computer will refuse to accept the data.

Student programmers operate teletypes and the CRT's to communicate directly with a computer. They may also use punched cards or paper tape to compile their programs on a variety of computers. The terminals at Johnson County Community College are capable of communicating with IBM Control Data, UNIVAC, or G.E. Honeywell computers. The access to a variety of large computers makes it possible to teach almost any major programming language. In addition, the students are not limited by computer size in the size and complexity of the programs they write.

Yes, they have no computer. Yet they have all the benefits of the very best in computer technology at a fraction of the cost. A non-traditional approach to data processing for the Johnson County Community College has yielded high performance at a minimum cost.

Ron Kohler

Provost, Raymond J.
Going Computer is Easier and Cheaper than You Think
School Management. Vol. XVI, No. 10 (October 1972), pp. 30-31

Small to medium-sized school districts with limited budgets are often hesitant about converting to computer use. Prescott High School in Arizona, in such a situation, eliminated at $15,000 data processing system and replaced it with a terminal tied into a DEC-10 time sharing system at nearby Yavapai Community College.

YCC loaned Prescott a terminal for several months after demonstrating its use in math and science classes. Thus encouraged, students, faculty, and administration took advantage of YCC's survey course in data processing. YCC's method was effective as Prescott leased a terminal helping defray computer costs at Yavapai.

Prescott has solved the administrative-educational use conflict by giving priority to instructional use during the day and doing administrative work in the evening. This schedule has allowed them to keep their total budget under $15,000.

Administrative advantages of the DEC-10 system and improvements over the previous data processing system have come about through the assumption of all data processing operations by YCC. This has led to greater convenience in scheduling, registration, grade reporting, and so forth. At the time this article was written, another terminal was in the planning stage.

This article would be most beneficial to high schools with a computer installation within a short distance. It appears that many of the advantages Prescott has realized would vary inversely with the distance of the terminal from the computer. Nevertheless, with the given conditions, one high school has found a relatively inexpensive way to computer facilities.

Stewart Weimer

Sutherland, Ivan E.
Computer Displays
Scientific American, January 1973
Pages 53-69

The field of computer science has many interesting and important subfields. One of these is computer graphics--the interactive use of TV-like computer output devices. Sutherland's article provides an excellent overview of the field of computer graphics, including some of the major problems and a histroical perspective on progress. The level of presentation is suitable for the intelligent layman.

There are two broad classes of computer-display systems now in common use: calligraphic displays and raster displays. Raster displays make pictures in the same way that television sets do, by painting the image in a fixed sequence from left to right and from top to bottom. Calligraphic displays "paint" the parts of a picture on the cathode ray tube in any sequence given by the computer. The calligraphic display has the advantage that information to be displayed can be stored in computer memory in any order. In the raster display information must first be sorted from top to bottom and from left to right so that it can be put on the screen in the correct sequence. One advantage of the raster display is that it requires less equipment and therefore is less expensive.

Two groups of users have been particularly interested in computer dispalys. One group has a pictoral problem in the workaday world for which it would like computer help. They may want to shape a metal part on a computer-controlled machine tool, for instance here one would describe the part to a general-purpose computer, which draws a picture of the part and verifies that the description is accurate. The second group of users is interested in gaining insight into complex natural or mathematical phenomena. For example, a physicist may program a computer to illustrate how elementary particles interact with their own electric fields, to give his students some feeling for quantum-mechanical behavior.

The author of this article has been involved in the development of high-speed displays that present an observer with a simulated three-dimensional environment. Two miniature cathode ray tubes are mounted on the user's head, one tube in front of each eye, so that the computer can control exactly what he sees. The position of the user's head is monitered and the computer gives a perspective picture appropriate to that viewing position. As the user turns his head the perspective picture changes just as if the object portrayed was in the room.

Ron Boys

Trocchi, Robert F.
So You're Going to Purchase a Computer System
School Management, Vol. 16, October 1972
Pages 32-33

At some point in the near future many, if not all, school administrations will be involved directly or indirectly with the selection of computer facilities for their schools. Even if professional consultants are involved, there are several questions one must consider.

1. What do you want the computer do do now and in the future? Computer services should be acquired on the basis of the school's needs.

2. What tasks can it do for administrators now and in the future? There are many uses of computers besides payroll, scheduling, student records, testing, grade reporting, inventories, budgets, and accounting.

3. Will the computer be used as an aid to instruction? Instructional use of computers is a rapidly growing field. Do you want a system which can be used for computer assisted instruction, problem solving or simulation? Such uses require a time-shared, interactive system, whereas administrative uses do not.

4. How much use will be made of the System? The type and amount of use will influence the size of the system you need.

5. Is a given machine's software (the programs and languages which make the system work) what you need? Different uses require types of software.

6. Will the system grow with you without unreasonable costs? A computer bought solely for administrative uses may be totally useless for instructional uses. The reverse is also true.

7. Are the benefits you will receive worth the costs? Keep in mind the computer's range in cost, from under $5000.00 to over $1,000,000.00. An inexpensive system may save you money now but cost you more in the long run. However, there is little value in buying more power than you can use within the next few years.

Although this list of questions is far from exhaustive, it does point up several ideas one must be alerted to in the selection of computer facilities.

David Dempster

Zaiden, Dennis J.
Insurance For Your EDP
School Management, (April 1973), pp. 26-28

EDP (electronic data processing) insurance provides coverage for such unique facets of automated data processing as the loss of computer programs and accumulated transaction data, and the consequences of an interruption in processing. If you do not have such coverage, it should be considered.

Your general insurance policies should be coordinated with the EDP policy. The EDP policy should tak up where the general policy leaves off so that there is no overlap in coverage. Fire and extended policies ordinarily insure against losses arising from fires, lightning, vandalism and so forth.

The first step in planning for EDP coverage is to prepare a complete list of all hazards. Next, place a dollar value on the estimated probable losses arising from these hazards. When this has been done, sit down with your insurance representative and compare the list with existing coverage in your present insurance policies. Deficiencies in coverage will then become apparent. All deficiencies cannot be covered, but you can cover most of them.

To receive the best possible coverage, the insurance representative should have prior experience in writing EDP policies. He should not be a novice to computer installations.

Various insurance policies can be written to cover all phases of operation in a facility. These may include hardware and software coverage, source material such as data stored in magnetic discs, expenses incurred during restoration of the operation following a loss, valuable papers and records, and accounts receivable. You may also obtain data processors liability to cover your employees.

Give special attention to intential acts such as theft, which is not specifically mentioned nor specifically excluded on the EDP policy. Also to be considered is liability to outside parties and insurance on services rendered by outside computer service.

Amount of coverage may range from actual cash value or replacement value to policies where you will absorb a part of the loss. A loss is determined by totaling the expenses of data processing after disaster occurs and deducting the normal cost before the disaster.

Though you may minimize your loss by other back-up measures, you must have insurance coverage to provide funds to restore your organization to normal operations. When you buy an insurance policy, you are mainly buying peace of mind.

Bill LeMaster

CHAPTER IX

COMPUTERS IN SPECIAL EDUCATION

Ray Dodson Mary-Beth Fafard David Malouf

SECTION A: OVERVIEW

by Ray Dodson and Mary-Beth Fafard

Special Education encompasses those education services made available to school aged handicapped children. The first category is entitled developmentally disabled which is defined as follows:

> ...a disability attributable to mental retardation, cerebral palsy, epilepsy, or other neurological handicapped conditions of an individual found to be closely related to mental retardation or to require treatment similar to that required by mentally retarded individuals, and: 1) the disability originates before such individual attains age 18; 2) and has continued or can be expected to continue indefinitely; 3) and constitutes a substantial handicap of such individual. (Developmental Services and Facilities Act, Public Law 91-517).

The second category is entitled learning disabilities. While there are many differing definitions of learning disabilities, the current definition included is a fairly representative sample of those existing in the field. Learning disabled children are referred to as exceptional in the following manner:

> An exceptional pupil is so labeled only for that segment of his school career (1) when his deviating physical or behavioral characteristics are of such a nature as to manifest a significant learning asset or disability for special education purposes; and therefore, (2) when, through trial provisions, it has been determined that he can make greater all-round adjustment and scholastic progress with direct or indirect special education services than he could with only a typical regular school program. (Dunn, 1973, p. 7).

The above definition implies that any child who can benefit from special services even though he isn't mentally retarded should be provided a special education. Frequently gifted children and the culturally disadvantaged fit the above definition and are included in special education, however, they have not been included in this chapter.

The overview will note current trends in special education and their implications for computers, as an educational tool; and look at unique administrative uses of computers in special education. Secion IX-B discusses the use of the computer as a diagnostic and evaluative tool in special education. In Section IX-C, reasons for utilizing computers as instructional tools are given. Examples of current computer programs are presented in Section IX-D. The final sections (IX-E and IX-F) contain bibliographic and reference materials.

Current Trends

Current trends in special education may eliminate the need for this unit in the handbook. If in the future all children become treated as exceptional individuals there will be less need for a separate special education program. An important trend to be considered is the legal mandate presently being handed down by the courts. The courts are saying that every child is entitled to an education irregardless of his handicapped condition. States are starting to respond by providing such services. Oregon has responded in the last year by instituting project "SCHOOL FIND", a program designed to identify all children currently not attending school or receiving an education (House Bill 2444, 1973 Oregon Legislature). By the summer of 1974 this staff will report their findings to the Board of Education in Salem which in turn will use the results to determine educational needs for these children in the future.

Another trend is currently referred to as "mainstreaming" which means placing as many special education students as possible back into regular classrooms. Isolation of special students from regular classes has created new problems in the social areas over and above the handicaps these youngsters already possess.

Mainstreaming has been the impetus for new educational arrangement in certain areas, or staffed with specialists who are available to help the entire school population with specific education has led to a new approach in teaching, which also has promise for regular classroom. This innovation is task analysis and criterion referenced evaluation. This operates through the establishment of instructional objectives broken into a hierarchy of skills towards which the student can work. Still another innovation quite consistent with task analysis is diagnostic teaching. Such teaching utilizes an on-going evaluation of progress, so immediate corrections can be instituted in the teaching.

A final trend in education and special education is the increasing utilization of paraprofessionals to not only relieve teachers of many tasks not necessary to instruction, but also to provide indigenous personnel in the schools to facilitate uniqueness and variety to instruction.

Two things should be noted about the foregoing trends. First, if they haven't already been adpted into the regular classroom, they certainly are amenable to use in a regular classroom. Secondly, the statewide use of a register to be maintained to see who is or is not receiving educational services. Where some handicapped children are so scattered as to almost make it impossible to deliver services in the traditional sense, they could be delivered via computer. Also, such a computer facility could be maintained as part of the resource room concept. Finally, task analysis, criterion referenced evaluations, and diagnostic teaching appear to be in an ideal position for computer application.

No attempt has been made to repeat material covered by other sections of this handbook. Special note should be made that the chapters on CMI, CAI and administrative uses of the computer are just as applicable to special education as they are to regular education. Two unique administrative uses of the computer in special education should be noted. The first difference is a need for a more co-ordinated effort at a state level to provide services to the handicapped. Already mentioned was Oregon's Project SCHOOL FIND to locate those individuals needing services. Maryland has instituted the Data System for the Handicapped (DSH) incorporating six Maryland State agencies to plan and provide coordinated and comprehensive services for such efforts because the recipients of special education services aren't concentrated in small geographical areas as is true for regular education. A second major difference is that the providion of services for the handicapped has created waiting lists which must be maintained and continually updated. The use of computers has greatly improved the use of waiting lists to plan for future services.

SECTION B

USING THE COMPUTER FOR DIAGNOSIS AND EVALUATION IN SPECIAL EDUCATION

by Ray Dodson

The logical sequence of events in a special education program is as follows:

1. Diagnosis of individual needs
2. Prescription of educational strategies
3. Behavorial objectives established
4. Individualized instruction
5. Evaluation according to objectives established

Such an educational program is desirable for all students but is most often found in special education classes at this time. Possibly computers will allow the expansion of such programs into new areas.

An essential ingredient in educational programs is that they are built on an adequate diagnosis of individual needs. Computers can potentially provide the kind of diagnosis needed to optimize programs for handicapped students. A good diagnosis should be based on an assessment that is both valid and reliable. Valid means the test measures that it purports to measure, while, reliability is an index of how consistent the measure is. Computers can provide a diagnosis that is highly reliable i.e., given the same information at different times, it will always arrive at the same conclusion. However, this leaves something to be desired, as it implies a static system where a dynamic system might be more appropriate. Brooks and Kleinmuntz (1974) report an interactive system where the computer program asks questions that anyone, such as parents or teachers, can answer after two weeks of being around the person being diagnosed. The answers to these yes-no questions are processed by the computer, which then prints out a list of 20 diagnoses with each having a probability assigned that it is correct. The dynamic part of this program is that if at a later date the correct diagnosis is known for sure, it can be fed back into the computer and the computer is updated. This feedback increases the validity of the assessment, but realiability as it is traditionally known would be lessened. However, the objectivity of such an approach causes this to not be a significant problem. The cost of operating such a program is supposed to be cheaper than paying a clinician for the same diagnosis, but even more important is the flexibility allowed for improving the validity of the assessment, not to mention the speed of doing the task.

Other efforts are already underway for putting intelligence tests on the computer. For example, a multiple-choice version of the vocabulary subtest for the WAIS has been developed. Such tests as this and the one mentioned in the previous paragraph make a diagnosis based on questions answered in an interactive system. Such a system generally precludes the use of other diagnostic tests. However, some programs currently in use take the results of a variety of diagnostic instruments and put them together for a much more complete profile on the child. Such an approach is quite sound in terms of current psychological test theory, as the standard recommendation to psychometricians is not to base a diagnostic decision on the results of one test or observation. Ideally many different sets of test results should be correlated into a decision, and the immenseness of such as task calls for a computer. Further enhancing the results of such an approach is that with the proper programming the computer can print out a completed psychological report based on the test data. This is very simply done by setting a variety of conditions to be present before a descriptive phrase or paragraph is printed out.

No diagnosis should lead to a single descriptive phrase or label for an individual student. Rather, each students' record or profile should show his strengths and weaknesses, as this is the essence of invidivualized instruction. For example, such a profile of strengths and weaknesses should not indicate low in math and high in reading; instead, the skills involved in math could be arranged in sequential order according to their difficulty. Once the hierarchial order is determined subtests could be devised on specific subjects and placed in a computer program. Upon completion of the subtests, a profile would be available for each individual student which would be capable of showing the presence or absence of such minute skills as proper use of the decimal points. (Pat Suppes drill and practice in arithmetic grades 1-6 is an example of this.) Such a capability would not only serve a diagnostic capacity, but an ongoing evaluation capacity as well.

Due to the storage and retrieval ability of a computer, it is now feasible to retrieve selected bits of information when needed. For example, the computer can be programmed to respond to "How many students haven't reached a specified criterion on a given subtest?" Such information is highly desirable as it can be used for grouping the students to get the optimum use of instruction. Such grouping provides the student an opportunity to practive his social skills in a group. Individualized instruction, if carried too far, does not provide adequate opportunity for the use of social skills. The diagnosed needs of various students, when put together by a computer, serve more functions than just determining ability grouping of special or regular classes. For example, such a composite of diagnosed needs could be used for a class evaluation, or as a way to determine the need for in-service training of the staff, etc.

Court rulings nationwide have emphasized the rights of the handicapped to an education. A major difficulty has been due to an inability to adequately assess the needs of the special student. As mentioned earlier, special education student are heterogeneous, but yet they also have some things in common. The problem is there are many different variables to be considered when planning educational programs for the handicapped learner. Dealing with so many variables using traditional planning techniques is expensive, so this creates a dilemma because of the cry for accountibility when public funds are spent. A way out of this dilemma might be through computer based diagnosis, evaluation, and prescription of programs. However, in order to make it more economically feasible, all student in a district should have such educational planning used with them. Only then can it be said that everyone has an equal opportunity to education.

Special educators are rather unique in that due to the lack of educational programs for the handicapped in regular school systems, they must seek employment in institutions where many handicapped youngsters currently reside. By virtue of the total care provided in an institution special educators are required to work with their wards in more than an educational setting. That is, they must be aware of the health and related needs of their students as well. The enormity of this responsibility becomes more obvious when you realize that mentally retarded youngsters have a shorter life span than does the general population. Computers have been especially helpful in an institutional setting where the admissions staff could feed in such information as IQ, age, diagnosis and handicap and have the computer help make decisions about the future placement of the child.

In summarizing this section, it is important to note that the computer provides a feasible way of improving education available to everyone, including the handicapped. Also, in every phase of education (i.e., diagnostics, prescription writing, establishing behavioral objectives, individualizing education and evaluating progress) computers can have a part to play. If indeed education for the handicapped is moving towards "mainstreaming", so that disabled learners don't need to leave the regular classroom for their instruction, it will get there quicker and do a more thorough job by using the computer when appropriate. Computers also have the potential of allowing regular students to be treated as "special" students.

Some space should be devoted to problems encountered when using computers or introducing them into a special education program. First of all is the cost justifiable for a small population? However, if the diagnostic and evaluation facilities are also made available to the average students, cost per student should represent much less of a problem. As mentioned earlier, it is feasible to do diagnostic evaluations cheaper with computers than hiring additional clinicians. Another problem is that some diagnostic programs are interactive with the client of the services; severely retarded individuals might not be able to interact with the computer without very close supervision. Another problem is that computer based diagnostic systems would likely still need a clinician to explain the results to some teachers or parents. However, fewer clinicians would be needed than are now traditionally used in support of special education services. Finally, it is said that diagnostic services should be conducted by someone who is able to provide judgement on a variety of variables to arrive at a conclusion.

But as it is typically done, the clinician usually bases his decision on the probability of certain events happening or conditions being present, and this could be done more objectively by a computer. Thus, the problems posed for the use of the computers as a diagnostic tool do not appear to be insurmountable.

SECTION C

COMPUTERS AS AN INSTRUCTIONAL TOOL IN SPECIAL EDUCATION

by Mary-Beth Fafard and David Malouf

Before a teacher decides to utilize a new method of instruction, a new mechanical device, or introduce new personnel into the classroom, the basic questions to ask are why should this be used and what effect will it have on the learning of children. Although these questions seem rather obvious, in the education world they are seldom asked. Education is noted for trying a variety of new "gimmicks" in a panacea fashion and only later asking its fundamental worth and effect on the learning of children.

When special education teachers are considering utilizing the computer with exceptional children, they must first ask: WHY?

WHY?

The educational characteristics of exceptional children suggest several advantages for utilizing the computer in presenting and planning instruction. Like programmed instruction which has been successfully used in special education, an instructional system using computers provides a learning environment which accomodates an exceptional child's needs in the following areas: (1) active learning with high student response rate, (2) immediate feed-back and reinforcement, (3) maintenance of attention, (4) individual pacing of instruction, and (5) infinite patience with sufficient repetition to insure learning. In addition to the advantages of traditional programmed instruction, computer assisted instruction in special education offers the possibilities of modifying content of instruction to meet the needs of individual exceptional children. It offers a medium by which a task that a student might consider demeaning would appear at a high level.

Cogen (1969) has listed 17 advantages of computer instruction in special education. They are provided as a detailed listing upon which special education teachers may draw in evaluating computer assisted instruction within their classroom.

1. The learner is given the opportunity to control the situation - a strategic position.
2. The student is required to participate in the learning process.
3. Individual instruction is provided for speed control, skipping and enriching.
4. Repetition or a new approach is possible where the child fails to comprehend an important concept.
5. Branching techniques support weak areas and by-pass unnecessary areas on an individualized basis.
6. Attention is engineered through sound and eye-catching devices.
7. Lessons may be changed before the pupil tires or loses interest, diminishing the undesirable effects of short attention spans and possibly helping to develop longer interest periods.
8. Computers could provide rest periods as needed for some individuals.
9. A multi-sensory approach which may help to alleviate the effects of impairment, is provided.
10. Pictorial perception may lead to knowledge and skills other than verbal.
11. Immediate tie-ins between sensory and symbolic aspects of learning are possible.
12. Extrinsic or intrinsic rewards may be used.
13. Immediate responses can be obtained where desirable, allowing for immediate adjustments and reinforcement.
14. The almost instant knowledge of results, at appropriate times, may serve as both motivational and corrective factors.
15. Students may benefit by being able to hear (and see?) themselves with resultant improvement in learning and self-images.
16. Structured programs in controlled situations help to make possible later applications of identified skills and knowledge (transfer).
17. Teachers may be able to tune in on individuals or groups and assist where necessary. (p. 40)

In Providence, Rhode Island, these advantages have been instrumental in planning a course program and workshop for teachers of the mentally retarded. (Computer Assisted Instruction, 1965). In the area of the deaf, the Stanford University Network has taken the salient features of CAI and applied time to mathematical instruction for the deaf in which statistical results indicated an advantage to using the computer. (Suppes, 1971). Another advantage pointed out by the New York League for the Hard of Hearing (1968) in a program utilizing a multi-sensory computer based program is the release of hearing and speech clinicians for more time in individual therapy.

Although this does not exhaust the reasons for using computers as instructional tools, the advantages noted are the most direct at getting at answering the "why" question. When one asks "why" they should reflect on "why not."

WHY NOT?

Although computers hold considerable promise in the education of exceptional children there are at present several limitations and drawbacks, some of which may be corrected by future improvements in computer hardware and software while others may in fact be inherent limitations for this approach to instruction.

1. A computer system is no better than its software (a fact previously mentioned in this book). At present special education is unable to explain or ameliorate the educational difficulties of many exceptional children. In this regard the computer shares the limitation of all other forms of instruction, which is our general lack of knowledge of learning.
2. The software is dependent upon the input from special education teachers who are involved with the programming of instruction for exceptional children. Until special education teachers get involved, canned and packaged programs may be as inappropriate to an excaptional child as a basal reader is to a child with reading problems.
3. Computer input is at present almost exclusively dependent on keyboards, with some auxiliary use of touch panels (PLATO) and light pens. While this may be an advantage with certain disabilities such as physically handicapped, it limits the auditory and kinesthetic responses which may be required with some children.
4. Computer output relies heavily on the printed word. This has many limitations for the problem reader, the mentally retarded and the blind, and the learning disabled who may experience particular reversal problems.
5. Certain types of disorders, i.e., behavioral, may not lend themselves to extended use of the terminal without direct supervision. Indeed with some of these children, the computer terminal may lend itself to increasing bizarre fantasies.

In summary, all these factors must be taken into consideration before deciding upon using computer facilities within a curriculum. Special education teachers must actively get involved in creating programs and wisely selecting materials for students.

SECTION D

SAMPLE PROGRAMS BEING USED IN SPECIAL EDUCATION

by Mary-Beth Fafard and David Malouf

MR. COMPUTER

The New York State Education Department's Division for Handicapped Children in contracting with the General Electric Research and Developmental Center has created language that can be used by special education teachers with only a half-hour's instruction, to design individualized daily lessons for special education students. The program, created by Dr. Philip Lewis and Walter Stone, has been used experimentally with populations of handicapped children in New York. Reports by teachers have been favorable and they point to the advantages in holding the student's attention and getting them interested in the curriculum. The typical lessons are as easy to construct as a daily lesson plan. A lesson may go as follows:

> "Hi. What's your first name?"
> The student would enter his name, "Bob".
> "I am happy to know you, Bob. I am the GE Computer. Today we are going to practice some addition: 5 + 2 = ?"

(Automated Education Letter, 1973)

The lesson would continue with correction procedures and statements of reinforcement and indicators of progress. The advantages of Mr. Computer are (1) it can be programmed for a wide-range of subjects, (2) it treats the student as an individual, but is impersonal, never sounding disappointed, (3) it's encouraging, (4) it's game-like, and (5) it provides for a doubling of the teacher interaction with students on an individual basis.

There are possibilities of using Mr. Computer in the home since it operates over television cable lines and the cost of the keyboard is fairly inexpensive. This provides a great advantage to home-bound exceptional children as well as children who need additional work on academic tasks at home.

CBRU

Computer Based Resource Units for special education are a service of the New York State Network of Special Education Instructional Materials Centers currently in operation on the eastern seaboard. This system was developed in order to aid the teacher in planning educational activities for handicapped children, to meet their individualized needs. Planning good resource units and individual activities for specific children can often take weeks, months, or years. Since it is contended that others are facing the same dilemma in all special classes, CBRU is "...seen as an attempt to help the searching teacher find the other who has already been there." (Gearheart, 1973, p. 138)

Each CBRU unit was developed by a group of teachers and specialists in an instructional area. The units contain the objectives, activities by which the objectives may be achieved, and evaluative criteria that may be used. All of this had been coded and stored in the computer. A teacher uses the CBRU Resource Guide in order to determine what units are available and how to fill out the form to obtain specific information on particular children. The teacher gets objectives that can be used for the entire group and also is able to check off variable characteristics for individual children. Once this is completed, it is fed into the computer which prints out ideas, suggestions of materials, content, activities, and evaluative tools for the entire class and work for specified individual children.

CBRU can only return information stored in its data banks, and it is presently an expensive service. However, at the moment the Bureau for the Education for the Handicapped, Office of Education, and The Regional Special Education Instructional Materials Center, State University College at Buffalo, New York provide this at no cost to teachers. (Gearheart, 1973)

DEBUSK CLINIC

The next program to be described was prepared by David Malouf, a doctoral student in special education at the University of Oregon. It directly shows how to implement a program with a select population of exceptional students.

PROJECT DESCRIPTION

This project was conducted at the University of Oregon during the spring quarter of 1974, using a computer terminal in the education building in conjunction with the instructional activities of the DeBusk Center housed in that same building. It was an informal project in the sense that no hard data were collected, nor were any experimental hypotheses formulated and tested. The purposes of the project included these:

1. To assess the feasibility of using interactive computer-presented instructional activities with learning disabled children.

2. To determine if and how teachers with little or no experience with computers could effectively incorporate computer-presented activities into their instructional programs.

The DeBusk Center is a special education practicum site under the supervision of skilled specialists in the remediation of learning disabilities. Each quarter, university students enrolled in the practicum are given experience tutoring learning disabled children who are referred by parents and schools in the Eugene area. The emphasis is on reading instruction based on a precision teaching model, and rigorously defined categories of exceptionality have been discarded in favor of attention to the children's individual patterns of strength and weakness. The quarter the project was conducted 15 children, ranging from elementary to junior high school ages, were served at the center by 11 student teachers and 3 supervisors (2 of whom had previous experience with computers.)

Given the individualized nature of the instruction at the DeBusk Center (no student teacher met with more than two children simultaneously), and the wide variety of student levels and needs, it was decided that the computer-presented activities should be dealt with as an education resource which the student teachers could use at their discretion, rather than as a system for presenting new material and determining curricular content and sequences. The latter course would require a rather large scale effort, which would be beyond the resources of this project, and which might diminish the value of the DeBusk practicum for the university students. Thus the programs were written to allow maximum flexibility, with controlled reading levels and provisions for input from the student teachers and the children.

The student teachers were introduced to the computer terminal, a model-33 teletypewriter, in a single demonstration session, and told that they were to be accompanied by one of the supervisors during the first few times they operated the system with their students. Since the terminal was also used by university students not associated with the practicum, a schedule was posted on the door of the TTY room to reserve specified times for the student teachers. A brief handbook was prepared and made available which detailed the procedures for logging on and off, and correcting errors, and which also contained a description and sample of each of the programs. The handbook, which was intended to be used in preliminary planning of lessons as well as during the actual operation of the machine, was not fully utilized during the quarter. Perhaps this was because it was easier for the student teachers to ask advice from one of the supervisors than to wade through the material in the handbook. However, poor utilization of the handbook seemed to reduce the effectiveness of the computer activities, and limited student teacher awareness of new programs as they were added. A more serious effort at computer assisted instruction would require an intensive program to familiarize the teachers with the equipment and the available programs, as well as to suggest possible applications of the programs to the remediation of specific student deficiencies.

The programs, which were written in BASIC by a graduate student with approximately 10 hours of course-work in computers, varied in length from 81 lines to 239 lines. They were completed one at a time during the first part of the quarter, and modified from time to time in response to feedback from the student teachers and supervisors.

The first program, GUESS, was a game similar to the one commonly known as "Hangman." Before the student sat down at the terminal, the student teacher was asked to input a word and to set the number of guesses the child should be allowed. The word was to be one previously encountered by the child, or one which conformed to phonic rules with which the child was familiar. When the child sat down and began the game, the rules were explained and then a series of blanks were printed, each of which represented a letter. The child was allowed to guess one letter at a time by pushing the appropriate teletype key. If the letter appeared in the word, it was inserted in the correct spot, and the machine printed a mild verbal reinforcer. If the letter appeared more than once in the word, the machine responded with a stronger verbal reinforcer ("Wowie, zowie..." which specified the number of spots in which the letter belonged. These verbal reinforcers were observed to be effective in maintaining student motivation at the game. This program was intended to be used in a variety of ways. The student teacher could, for example, supply the child with an initial, final, or medial letter, a letter cluster, a definition of the word, a synonym or antonym, etc. Thus the exercise could be used to practice a variety of previous learnings. However, poor student teacher awareness of the possibilities of the program limited its use, and words were sometimes entered with which the child was unfamiliar or which used phonic rules which the child had not mastered.

The second program, SUPER, (see Appendix A) was a rather long story which incorporated personal information about the child into the narrative. The student was first asked to enter such data as his/her age, sex, school, teacher's name, a friend's name, etc. A correction routine followed which allowed the child to change or correct the answers before the story was printed. Only after the child had indicated that the answers were correct was the story printed. Student response to the story was quite enthusiastic. It was reported that several of the students took the story to school, and feedback from their teachers at school was also positive.

The third program, SCARY, was a shorter story which could be repeated several times in one session, and which allowed for greater student input, for example a description which could take up to a line of the text. Student reaction to this story was again observed to be rather positive, causing one child to fall of his bike with exuberance.

In conclusion, as was previously mentioned, this project produced no hard data by which to judge the educational value of the computer-presented activities. In a situation where the children were receiving highly individualized and intensive instruction, the computer assumed the role of a motivator and positive reinforcer. Several of the student teachers in which computer time was used as a reward contingent upon the accumulation of a number of points or tokens for desired behavior during the turorial sessions. Children at the terminal were required to do a substantial amount of reading, for example questions to be answered, rules to be read, and stories in which they were the heroes. Also, the students were required to do some typing of letters, words and even phrases. Thus the time they spend at the terminal involved considerable practice of reading and writing skills, and presented these exercises in an enjoyable way.

It was also observed that the student teachers became interested in the possibilities of computers in education, and several asked for more information on the subject. A greater degree of utilization of the computer-presented activities can probably be achieved by a more extensive program for training the student teachers in operating the computer and adapting the activities to their individual instructional requirements.

Appendix A

SUPER 13:27 20-MAY-74

```
Hi, kid.  What is your first name?  (Just your first name.)
 ?Ron
Hello, Ron.  I am going to write a story about you.
But first, I have to ask you some questions.
Answer them carefuly.  Ok?  Here we go!
What is your last name, Ron?
 ?Jones
Ron Jones.  That is a nice name.
How old are you, Ron?  (Use the number keys on the top row.)
 ?11
Pardon my asking, but are you a boy or a girl.  I can't see you.
 ?A boy
What is the name of the city you live in, Ron?
 ?Eugene
```

What school do you go to, Ron?
?Patterson
What is the name of your teacher at Patterson?
?Miss Fafard
Tell me the first name of one of your good friends, Ron.
(Just the first name.)
?Dave
What is Dave's last name?
?Andrews
What is your favorite color, Ron?
?Yellow

Here are the answers I have:

1- Your name is Ron Jones.
2- You are 11 years old.
3- You live in Eugene and go to Patterson School.
4- Your teacher's name is Miss Fafard.
5- You have a friend named Dave Andrews.
6- Your favorite color if yellow.

Do you want to change any of these answers before I print the story?
?Yes
Type the number of the line that you want to change.
If you want to change more than one, choose just one now.
?5
What is your friend's first name. Just the first name.
?Dave
What is Dave's last name?
?Apple
Any more corrections?
?No
Ok. Here is the story about you. I hope you like it, Ron.

Ron and the Super Belt

By Ron Jones
and the computer

This is a story about Ron Jones. He was 11 years old. One day, as Ron was going home from Patterson School, he saw a belt in the ditch. The belt was yellow and it had a button on it. Ron put on the belt. It fit him well. Then he hit the button.
*****WHAM!*****BAM!*****SLAM!*****KABLAM!*****
Ron became super boy!

Ron jumped up and flew away. He flew higher and higher.

Higher than the trees. Higher than the clouds. He looked down and saw his friend, Dave Apple playing in the street. Oh gosh! A car was coming fast! Dave didn't see the car! Ron flew down and picked Dave up just as the car went by.

'Thank you,' said Dave. 'You saved my life.' Then Dave looked at Ron and asked, 'Who are you?'

'Don't you know me?' said Ron. 'I am your friend, Ron.'

'No, you're not Ron,' said Dave. 'Ron doesn't have a yellow belt, or a yellow cape. Ron can't fly.'

'Then I am super boy,' said Ron. And he flew away.

Well, that day Ron had a lot of fun. He flew so high that Eugene looked like an ant hill. He played tag with an eagle and a hawk. Then he flew to Patterson School and got all the balls off the roof. He threw the balls to the kids who were playing at the school. They all yelled, 'Thank you,' but they didn't know who he was.

Then Ron saw Miss Fafard, his teacher. Miss Fafard had a big stack of books and was walking to the bus stop. Ron flew down and got all the books, then he flew up high. So high that Miss Fafard looked like a little dot, a little spot.

'Bring back those books, Ron!' yelled Miss Fafard. 'I know who you are, Ron, and I want my books!'

'What did you say?' yelled Ron.

'I said bring back my books, Ron, and stop messing around!'

Ron flew down and landed by Miss Fafard. 'How did you know it was me?' he asked. 'All the kids think I am super boy because I can do super things. They don't know who I am.'

'Oh, Ron,' said Miss Fafard, 'I have always felt you are super. I have always felt you can do super things.'

Ron took off the belt. He handed the books and the belt to Miss Fafard. 'Here are your books,' he said, 'and here is my super belt. I am sick of flying.'

'Thank you,' said Miss Fafard. 'You do not need a super belt to be super, Ron.'

The next day, Miss Fafard wasn't at school. 'Is Miss Fafard mad at me or sick?' asked Ron. Then, after school, as Ron was going home, he heard a voice from way up high.

'Hello, Ron,' said the voice. Ron looked up and saw Miss Fafard flying above him.

'Hello, Ron,' said Miss Fafard, 'Thank you for the super belt. I am now super teacher. Do all your home work and don't chew gum in class. I am going to Washington to be President! Goodbye, Ron, goodbye.....' and Miss Fafard flew away.

'So Miss Fafard wants to be President,' said Ron. "Adults just don't know how to have fun.' And he ran to Dave's house to play.

THE END

BIBLIOGRAPHY

1. Brooks, R. and Kleinmuntz, B. Design of an intelligent computer psychodiagnostician, Behavior Science, v 19, n 1, (1974) pp. 16-20.

2. Cartwright, G. Phillip and Cartwright, Carol A., Early identification of handicapped children: A CAI Course, Journal of Teacher Education, v 24, n 2, (Summer 1973) pp. 128-134.

3. Cogen, V., The computer's role in education and Use with the exceptional child, Mental Retardation, v 7, n 4, (1969) pp. 38-41.

4. Dunn, L. M., Exceptional Children in the Schools, New York: Holt, Rinehart and Winston, Inc., (1973).

5. Evans, Rosemary and Simkins, Katherine, Computer Assisted Instruccion for the Blind, Education of the Visually Handicapped, v 4, n 3, (Oct. 1973) pp. 83-85.

6. Gearheart, B. R., Learning Disabilities Educational Strategies, St. Louis, C. V. Mosby Company, (1973).

7. Hathaway, Starke R., MMPI and the Computer, Psychologie, v 31, n 4, (1972) pp. 277-280.

8. Kleinmuntz, B. and McLean, R. S., Computers in behavioral sciences: Diagnostic interviewing by digital computer, Behavorial Science, v 13 (1968) pp. 75-78.

9. Knutson, Jack M. and Prochnow, Robert R., Computer Assisted Instruction for vocational Rehabilitation of the Mentally Retarded, University of Texas, (Aug. 1970).

10. Levy, Joel and Finder, Stan, Computer Assisted Management of State School Waiting Lists and Admission Procedures, Mental Retardation, v 9, n 5, (Oct. 1971) pp. 30-34.

11. Schwartz, Barry J. and Others, An Accident Reporting System: A Tool for the Institutional Administrator, Mental Retardation, v 11, n 3, (June 73), pp. 22-25.

12. Suppes, Patrick, Computer Assisted Instruction for Deaf Students, American Annals of the Deaf, v 116, n 5, (Oct. 1971) pp. 500-508.

13. -----"Mr. Computer" Design Individualized Lessons for Handicapped Students, Automated Education Letter, v 8, n 6, (June 1973).

14. Computers are for teaching, New York League for the Hard of Hearing, New York, (1968).

15. Computer Assisted Instruction for the Mentally Retarded, Providence College, R. I. (Aug 1965).

16. Computer Assisted Planning: A User Manual for Computer Based Resource Units, SUNY - Buffalo, (Oct. 1972).

17. Guidelines for developing Computer Based Resource Units, SUNY - Buffalo, (Mar 1972).

Atkins, Martin and William Holloway
Computer Assisted Individual Student Programming
Journal of Learning Disabilities V. 4, n 8 (October 1971) pp. 46-51

Stating that quality education can become a reality by implementing individualized instruction for all students, the authors present a Student Program Management system approach for individually tailored learning prescriptions. The program was tested in Bridgeport, Michigan on handicapped children. It is "essentially an educational computerized tracking system which keeps tab, intervenes, and corrects a course of action" on students recognized to have learning problems. The authors see the program not as a panacea, but as an aid in setting up and maintaining accurate educational prescriptions for special children. There is a coordinator with the SPM who translates reports from reporting personnel into computer language. The program not only stores information on each student, but prints out prescriptive data for the educational planning.

This was an interesting article in that it pointed to one of the ways a computer could be used in Special Education programs. I would have liked to see more information on how the prescriptions were developed for the computer and some follow-up studies on the computers' accuracy.

Mary - Beth Fafard

Donohue, James F.
Computer-Based Study of Mental Retardation
Computers and Automation V. 18, n 11, (October 1969) pp. 50-52

The study and treatment of mental retardation have made great advances in recent years. One tool helping to further sophisticate these advances has been the computer. This article is based on the work at Pacific State Hospital (PSH) for the mentally retarded at Pomona, California.

While people tend to confuse mental illness and mental retardation, they are two very different disabilities. Furthermore, mental retardation is as heterogeneous as other disabilities and thus some skill is required at making a good diagnosis of the actual problem. PSH incorporated a Honeywell Model 1200 computer to assist in their diagnostic work. With the aid of this computer, the staff was able to get a profile of those patients dying while at PSH (the mentally retarded do have a shorter life span). The patient's probability of dying while in the hospital was directly related to his/her age, IQ, diagnosis and handicaps. Once this profile was known, the admissions staff could get such data during the patient's first day at PSH and forward it to the chief physician where appropriate specialized medical attention could be provided. The incidence of death for the high-risk group dropped from 13 per cent in 1962 to 2 per cent in 1969. PSH feels the same techniques could also be applied to hospitals for other than the mentally retarded.

The success obtained in lowering the death rate caused PSH to look for other computer applications for treatment of the retarded. The data already available was also found to be useful in predicting the incidence and prevalence of infections and other diseases. As patients with certain profiles were admitted, specific medications were immediately administered. For example, where dysentery was once common at PSH, it has been all but wiped out.

Another use for the computer has been in the taking of EEG's. The EEG lacks sophistication because of "noise" caused by cerebrospinal fluid surrounding the brain. This fluid causes a sort of static which distorts the EEG reading. However, the computer, through an averaging process, is able to greatly improve the accuracy of the reading. This same technique is also used to read other responses evoked from the brain. This has enabled the staff to do a better job in evaluating the patients' sight and hearing.

Finally, PSH has taken its research into communities in order to better study mental retardation. The findings reveal that a person's socio-economic background and race are frequently overlooked when trying to determine the causes of retardation. They found that when a poor person from a minority group had the same characteristics as a middle class caucasian person, only the poor minority group person was generally categorized as "retarded." Such findings are causing communities to look at why they label some people mentally retarded. The communities can then revamp the delivery of their services in terms of the answers to these questions.

The computer was also found useful in helping to set up alternatives to hospital placement or special education classrooms.

Ray Dodson

Horn, Lester W. and Gleason, Gary M.
Teaching a Unit on the Computer to Academically Talented Elementary School Children
Arithmetic Teacher, V. 17, (1970) pp. 216-219

In 1969 an experimental project designed to teach concepts of the computer to 75 academically talented fifth graders and sixth graders was conducted at Pensacola, Florida. Data processing instructors at the college taught the computer concepts unit. The children received a one and one half hour period of instruction for ten consecutive weeks. They covered the areas of the history of computers, numeration systems, computer hardware and FORTRAN programming. The culminating activity involved the students in writing and running a short FORTRAN program. Technical as well as general aspects of the computer were covered.

As a technique to convey what happens at each step in the program, the class acted as a computer. "Each child was assigned a FORTRAN statement and executed his single instruction as the computer would." This was especially effective for demonstration of branching on the basis of an "IF" statement. The students also wrote two programs by themselves. Time in class was used to correct errors and to rerun student programs.

While the project was deemed an overall success, it was felt that better results could have been achieved if more time had been available for instruction. It was further determined that coordination of the computer studies with topics studied in mathematics and science will become necessary.

Speculation was also made about the kinds of techniques used to teach the computer course. It was suggested that students with average ability may need different methods of instruction than the gifted children.

Kathleen M. McMannan

Harman, C.F. and Raymond, C.S.
Computer Prediction of Chronic Psychiatric Patients
Journal of Nervous and Mental Disease (1970) pp. 490-503

Reports on the use of computers to perform the multivariate analysis to be used for prediction of successful rehabilitation of chronic psychiatric patients. The program was rated as successful in that patient clusters were discovered which have extremely high correlation with the positive prognosis.

As the author suggests, the extensive data already exists to make great improvements in prognostic ability. However, there is so much information available, and handling the multiple correlations is so difficult to do by hand, that only by introduction of the computer, can the task be made manageable. This could be a significant step in the provision of improved diagnostic procedures for the seriously psychiatrically impaired.

Cogen, Victor
The Computers' Role in Education and Use With the Exceptional Child
Mental Retardation V. 7, n 4, (1969) pp. 38-41

Stating that "....technological advancement refuses to wait for the educational lag"........the author examines the operations of the computer, its role as an assist to the teacher and instructional process, and its advantages to special education. He notes that what is to be avoided is the flooding of the educational forum with "programless teaching machines". A description of CAI is given and the wide range of possibilities that it has for the educator of handicapped children. The author lists the following: 1) examples of instructional programs for technology-based education, 2) advantages of computerized instruction for the mentally handicapped, 3) utilization of information retrieval systems for the handicapped. Also, the author elaborates on the data processing possibilities and personnel training programs that are available with the computer.

This is a must article for people in special education. It thoughtfully gives some of the advantages of the computer and its possible uses. Special educators shall have to implement many of these ideas; this article serves as a possible beginning.

Mary Beth Fafard

CHAPTER 10
SURVEYS

The material contained in this chapter consists of abstracts of various surveys. Each abstract contains the name of the survey, a summary of its content, and a reference. Those interested in the survey can obtain a copy by writing to the person or agency indicated in the reference. THE ABSTRACTS REPRESENT ONLY PARTIAL RESULTS, and are provided as an index to surveys that might be of further use to you.

CHAPTER X.1

SURVEYS

SECTION A: ABSTRACTS OF SURVEYS

Computers in Higher Education
Report of the President's Science Advisory Committee; 1967
Washington, D.C., 79 p.

This report was divided into 5 chapters with 11 appendicies. The chapters are:

I. Introduction
II. Computers and Undergraduate Education
III. The Computer Science Student
IV. Interaction Between Research and Educational Uses of Computers
V. The Computer and Secondary Education

The appendicies are:
A. Computers in Education
B. Some Facts of Life About Computers
C. Computer Languages
D. Educating the Faculty in Use of the Computer
E. The Large University Computational Facility
F. What Computer Facilities are Appropriate
G. Estimation of Required Computer Capacity and Cost
H. The Communication Problem
I. The Growth of the Computer Industry
J. Examples of the Use of Computers in Course Work
K. Statements Concerning Value of Computing

Even though the report is relatively out of date (1967) many of the findings would still be appropriate if the study were made today. Two main themes run through the study. The first is that it is certianly in the national interest to have adequate computing for educational use in all institutions of higher education by 1971-72, and the second theme is that the cost of doing this, although large in total dollar amount, is not large when expressed on a cost per student basis. This cost is estimated to be $60 per student (this is comparable to the $50 to $200 per student per year for college libraries and a $95 per chemistry student per year for one chemistry laboratory course in a 4 year college).

The panel on computers in higher education made the following ten major findings and recommendations. These ten are broad summaries of the specific information in the report.

1. Approximately 35 percent of college undergraduates are enrolled in curricula in which they make valuable use of computers in a substantial fraction of their courses. An additional 40 percent are in curricula for which introductory computing training would be very useful, and limited computer use should be part of several courses. The remaining 25 percent could make some use of computers in one or more courses during their college education, but computer training is not now important to their major studies.

In 1965 less than 5 percent of the total college enrollment, all located at a relatively few favored schools, had access to computing service adequate for these educational needs. However, it is practical to supply adequate computing service to nearly all colleges by around 1971-72.

We recommend that colleges and universities in cooperation with the

Federal Government take steps to provide all students needing such facilities with computing service at least comparable in quality to that now available at the more pioneering schools.

2. One of the major problems in providing the necessary educational computing is the cost. The yearly cost of providing this service will rise to a total (for baccalaureate programs and 2-year colleges) of about $400 million per year in 1971-72 in addition to the relatively smaller costs required for faculty training and associated research. It is beyond the capabilities of our colleges and universities to bear all of this cost in this time period.

We recommend that colleges be encouraged to provide adequate computing through government sharing of the cost. Such governmental cost sharing should include special grants to cover transient costs when service is being initiated or larger facilities are being installed. It should also provide a portion of the annual cost of continuing service.

3. Government accounting practices have made it very difficult for colleges and universities to utilize fully that Federal and private support for computers or computer service intended for unsponsored research and education (as distinguished from research paid for by grants and contracts).

Treatment for a grant for educational use of a computer at a reduction in total cost reduces the hourly charge for computer time paid by all users and has the effect of shifting research costs to educational users. The Department of Defense has recognized this and now has an agreement with the National Science Foundation not to treat NSF educational grants for operating expenses as a reduction in sponsored research costs.

Many schools cannot now afford to pay for educational and unsponsored research use of computers by students and faculty even though there is time available on their computers. Consequently, some college and university computers now available for educational and unsponsored research use are standing idle for major portions of the operating week.

We recommend that the present DOD-NSF agreement be extended to other government agencies and private supporters and include both capital and operating cost grants. Additional Federal funds should be made available immediately for support of computing service used for education and unsponsored research activities at institutions presently having the required facilities.

4. We find that any expansion of the educational use of computing depends heavily on increased knowledge of computing by faculty in most disciplines. Such knowledge usually can be provided by intensive 2 to 6 week periods of faculty education. The extensive activity of the National Science Foundation in sponsoring summer institutes provides a useful model.

We recommend an expanded faculty training program to provide adequate faculty competence in the use of computing in various disciplines.

5. There is a great need for specialists trained in the computer sciences at the bachelor's, master's, and doctorate level. The whole success of

educational computing and continued improvement in its use depends on expanded education and research in computer sciences. This education requires a good faculty and access to very good computing facilities both for course work and research.

We recommend that the Federal Government expand its support of both research and education in computer sciences.

6. The cost of computing is a continuing expense, like light or water, rather than a capital investment, like the initial cost of buildings.

We recommend that the Government agencies which support computing allow the schools to be free to apply the funds either to the purchase or rental of equipment and the support of staff, or to the purchase of service.

7. The optimum mechanism for providing computers will differ from campus to campus. However, in many cases it appears economical and effective to supply adequate and dependable service from large computing centers.

We recommend that universities and the Government cooperate in the immediate establishment of large central educational computing facilities capable of serving several institutions.

8. Because of inconsistent Government and university accounting practices, the great variety of sources of computing support, and the experimental nature of computer use, some universities have had difficulty in determining and controlling their computer costs. Informed decisions regarding expansion and/or budgeting for current operations cannot be made without accurate cost information. Errors made at this stage can only lead to the diversion and dissipation of university resources needed for other educational purposes.

We recommend that universities and colleges develop and use accounting procedures which accurately measure the cost and utilization of computer services. With such information the allocation of computer time for research and education and the anticipation of associated costs should be placed on a realistic and measurable basis.

9. Proper introduction of computing into secondary education is desirable and growing. Not enough is known about the best ways for introducing computing and we were not able to consider this adequately in the time available.

We recommend that NSF and the Office of Education jointly establish a group which is competent to investigate the use of computers in secondary schools and to give the schools access to past and present experience. Cooperation between secondary schools and universities, and particularly providing service to secondary schools from university centers, should be encouraged.

10. There is inadequate information about the number and level of skills of personnel now employed in the field of computers, and there are no meaningful forecasts.

We recommend that the Federal Government collect meaningful data concerning computers and the jobs, personnel, and educational facilties associated with them, and endeavor to make useful annual forecasts.

[illegible], Linda; Moursund, David, and Whitney, Rusty

Survey of Computer Science Education of the Teachers of Computer Science in METCOM

Spring 1973

During the 1972-73 school year a survey was taken in the Portland Metropolitan area to ascertain the teacher education level of the computer science coordinators of the METCOM consortium project. This consortium has two Hewlett-Packard 2000 series computers to supply the three county (Clackamas, Multnomah, and Washington) metropolitan area high schools with computer resources. This survey was conducted to provide data to the computer component of the Oregon Mathematics Education Council (OMEC). The following is a summary of the 18 respondents answers.

1. What computer classes have you taught?

- [illegible] Computer Science or Introduction to Computers
- 8 Computer Mathematics or Problem Solving
- 2 Computer integrated into regular math
- 3 Computer Programming

Of the 18 teachers responding to this question, 4 teach no regular computer classes this year. Ten teach one or more sections of one class. Five teach more than one class.

1a. What teacher workshops have you taught?

- 13 None
- [illegible] [illegible] at some time
- 3 Have taught more than one workshop

Of workshops taught, 2 were offered by PSU-DCE and 2 by community colleges. Two teachers have taught only district inservice and two have taught classes offered by other institutions.

2. Have you developed curricular materials?

- 4 None
- 14 Have developed something.

Of the materials developed, only two teachers have developed materials that are used in districts other than their own.

2a. Have you worked in computer-related jobs besides teaching?

- 16 No computer job.
- [illegible] College research assistant, computer related.
- 1 Industrial computer position (2 years).

3. Do you do other computer-related teaching?

- [illegible] No other teaching
- [illegible] Sponsor computer clubs or other student organizations which use computers
- [illegible] Informal tutoring or teaching
- [illegible] No response

4. What classes should be offered teachers?

Class	No. requesting
1. Computer class teaching methods	3
2. Integration of computer into regular curriculum	9
3. Beginning Programming	4
4. Intermediate or advanced programming	3
5. What computer science is about, effects of computer on society, computer literacy	3
6. General data processing	1
7. Assembler language	1
8. Colorado Project type	2

5. What equipment do you have access to?

No. of Terminals	No. of responses
0	1
1	5
2 (Usually 1 off-1 on line)	10
More than 2	2

Percent of time equipment is fully used	No. of responses
100	6
75-100	2
50-75	5
25-50	3
less than 25	2

These people all have access to Multnomah County's HP except for the one person with no terminal. Four also use programmable calculators.

6. What languages can you use?

Language	No. of responses
BASIC	18
Fortran (II or IV)	15
FOCAL	1
COBAL	1
Assembler or machine language of some type	2

No. of languages known	No. of responses
Only 1	3
Knowing 2	10
Knowing more than 2	5

7. What computer classes have you taken?

No. of classes	No. of responses
0	1
1	5
2 or more	12

NSF Institutes Taken	No.
1 of	4
Other	2

All of the people who were in institutes are in the 2 or more group above.

3. What college work have you done?

All but one teacher have bachelors degrees in mathematics of which thirteen received them from an Oregon school.

Thirteen teachers have earned a masters degree in math or math ed, of which ten received the advanced degree from an Oregon school.

In addition, two of the teachers have advanced degrees in disciplines other than mathematics.

Darby, Charles A., Korotkin, Arthur L., and Romashko, Tania
Survey of Computing Activities in Secondary Schools
American Institutes for Research, October 1970, 153 p

A mail survey of 23,033 public secondary schools in the continental United States was conducted during the period January to June of 1970 to gather information regarding primarily, instructional use of computers. In addition, interviews were conducted at 90 selected schools throughout the country, including six in Oregon.

The study concluded that the use of computers, especially in instruction in secondary schools, has grown rapidly, However, the diversity of use is still limited. The most prevalent applications are problem solving and EDP skills. This is a bit misleading in that teaching computer programming is considered an EDP skills training course, even though the course is not intended to produce production programmers and is intended to be part of a vocational program. The major emphasis of computer application is on teaching students to use a computer as a tool of learning more about the subject area in which the computer is being applied. Most instructional applications occur in the mathematics courses. Local sources provide the majority of funds for instructional computer use. Plans for future use generally call for expansion of present applications.

Darr, [illegible]

1973 Directory of Computing Facilities in Institutions of Higher Education Throughout North America

Prepared in conjunction with the Annual Seminar for Directors of Academic Computing Services, University of Colorado, 10 August 1973

In a survey of 1,474 institutions the author received replies from 810 schools. This report provides the following information on academic computing facilities in North American higher education.

- Type of Control--State, Local, Federal, Independent
- Primary Service Function--Instruction, Research, Administration
- Type of I/O service--Local Batch, Remote Batch, Time Share
- Enrollment
- Institution Budget
- Computing Center Budget
- Type and model of Computer

In addition the person responsible for academic computing is identified along with the respective title, address and phone numbers.

Richard V. deGrasse
<u>Remote Computing in Higher Education: Prospects for the Future</u>
Study on Teleprocessing Networks in Higher Education
1971, 102 p

The following is a summary of the answers to the three questions which formed the original basis for this study, as well as a list of the important findings.

<u>Question 1: Is a national educational teleprocessing network evolving by itself?</u>

No. Not if a national network is defined to be the connecting of the majority of institutions of higher education on a single network using a common utility. There are approximately 25 small disconnected regional networks. There is very little incentive for them to connect together. The future will bring more of these small local networks into being. To develop a large national network will require a great deal of government stimulation.

<u>Question 2: Is a national educational teleprocessing network worthwhile?</u>

Yes. Providing it embrases as many as possible of the 1250 institutions which presently do not have computing facilities.

<u>Question 3: What is the effect of computer hardware and software?</u>

Hardware developments will have very little effect on the evolution of remote computing. The reason is that presently the majority of institutions have very little, if any, hardware in the first place. Software will be the most difficult. Programs which will serve a large number of institutions will take time to prepare.

<u>Findings</u>

1. There must be a more equitable distribution of computing services in higher education.

2. Computing is a small part of higher education (2.5% to 3% of expenditures) and will evolve with changing patterns of higher education and will not cause any drastic changes in higher education.

3. The three uses of computing in higher education, instruction, research, and adminiscration, are at different stages of development. Instructional use is the newest and could benefit the most from networks.

4. Research users will find use from networks providing government support is available.

5. Administrative use of networks is unlikely under the present structure of higher education.

6. Networks centered around large, mainly public, institutions, will

serve as a means of involving small institutions in computing.

7. Networks will falter in the long run (5-10 years) unless there is a change in the organizational structure of higher education or unless a separate entity is created involving all of the institutions using the facility.

8. A more equitable means of distributing communications costs must be worked out.

9. Data communications is 1-2% of computing activity in HE. Therefore, tariff changes would have little effect on the development of networks.

10. The greatest potential for reducing the costs of data communications lies outside the common carrier.

11. Although the number of terminals used by HE will increase, the percentage of the total number of terminals used nationwide will decrease. By 1980, HE will account for 12,500 of the 2.5 million terminals in the country.

12. Data communications problems in HE do not involve high speed long distance, multichannel communications systems.

13. Alternative data communications networks besides the common carrier are nearly non-existent.

14. More sustained federal support is needed for CAI so it does not fall in the same trap as ETV, too much emphasis on technology. Federal support seems to disappear once the technology is perfected.

15. Present NSF policy of encouraging small colleges to utilize computing for instruction in all disciplines appears correct.

Hamblen, John W.
Inventory of Computers in U.S. Higher Education, 1969-70
Southern Regional Education Board, March, 1972,
Atlanta, 396 p

This study, done in 1969-70, is the third of three studies done by John W. Hamblen. The first was done in 1964-65, and the second in 1966-67. Many of the tables and graphs compare all three studies to show patterns. The inventory was organized by computer installation rather than by institution. That is, if there were three different installations within one institution, three separate returns were filed by that institution.

The returns of the institutions were stratified by three major characteristics; control, highest degree offered, and enrollment (based on Fall 1968). The breakdown is shown below:

Control:
1. Public
2. Private

Highest Degree Offered:
1. Associate
2. Bachelor's
3. Master's
4. Doctorate

Enrollment:
1. Below 500
2. 500 - 2,499
3. 2,500 - 9,999
4. 10,000 - 19,999
5. 20,000 - and above

The survey covered four broad areas: computer systems, expenditures, and sources of support, computer installations by type of usage and title of officer, and computer sciences and related degree programs.

Computer Systems

Several facts regarding the types and numbers of computers and computing systems were brought out by this survey.

1. There were by far more IBM computers than any other type. IBM accounted for 63% of the computers while the second most popular manufacturer, DEC, accounted for 12%. The remaining installations were divided among approximately 25 different manufacturers.

2. Public installations reported almost fifty percent more computers than private institutions, even though the number of institutions was approximately equal.

3. There was a high correlation between the enrollment of the institutions and the number of institutions having computing facilities. This same high correlation existed between types of degrees granted and computing facilities available. That is, the higher the degree granted (associate,

bachelor's, master's, doctorate) the more institutions having computers.

4. The percent of institutions having computing facilities went up with each of the three inventories conducted, in the following manner: the 1964-65 inventory showed [illegible]% of the institutions had computers, the 1966-67 showed [illegible]%, and the 1969-70 showed 59%. The growth was not linear.

5. The number of students who did not have access to a computer dropped from 2.4 million in 1964-65 to .8 million in 1969-70.

Expenditures and Sources of Support

The trends indicated by this section were:

1. The 2807 institutions responding reported spending $472 million on computer related activities. The source of this money was:

   a. the institution--$331 million - 70%
   b. all federal support - $79 million - 17%
   c. other - $62 million - 13%

2. The amount of money spent by public institutions was over two times the amount spent by private institutions.

3. Institutions granting doctorates spent over twice the amount of money as the sum of all those granting associate, bachelor's, or master's degrees, even though the institutions granting doctorate degrees accounted for only about one-eighth of the total number of institutions.

4. Of the total expenditures, approximately 30% was for on campus instructional use, 32% was for research, and 34% was for administrative use. The remaining 4% was categorized as miscellaneous.

5. By using estimated total expenditures for each of the fiscal years from 196[illegible] through 1968, and fitting the data to several different models, the equation of the curve of best fit turned out to be linear (Y=-56+59.6X). Table III-12 shows estimated expenditures through 1980.

Computer Installations by Type of Usage and Title of Officer to Whom Head of Installation Reports

[illegible]% of the installations reported the usage of the computer was spread among all three categories--administrative, research and instruction. Of the [illegible] computer installations responding, 793 reported this type of usage. This amounts to [illegible]%. The next largest category was research and instructional use, with no administrative use. There were 202 installations reporting this kind of usage.

Computer Sciences and Related Degree Programs

Much of the information in this area is summarized in figure V-1. By using the author's estimates, by academic year 1974-75, the institutes of higher education will be producing 500 Doctorates, 3500 Master's, 8000 Bachelor's, and 10,000 Associate degrees.

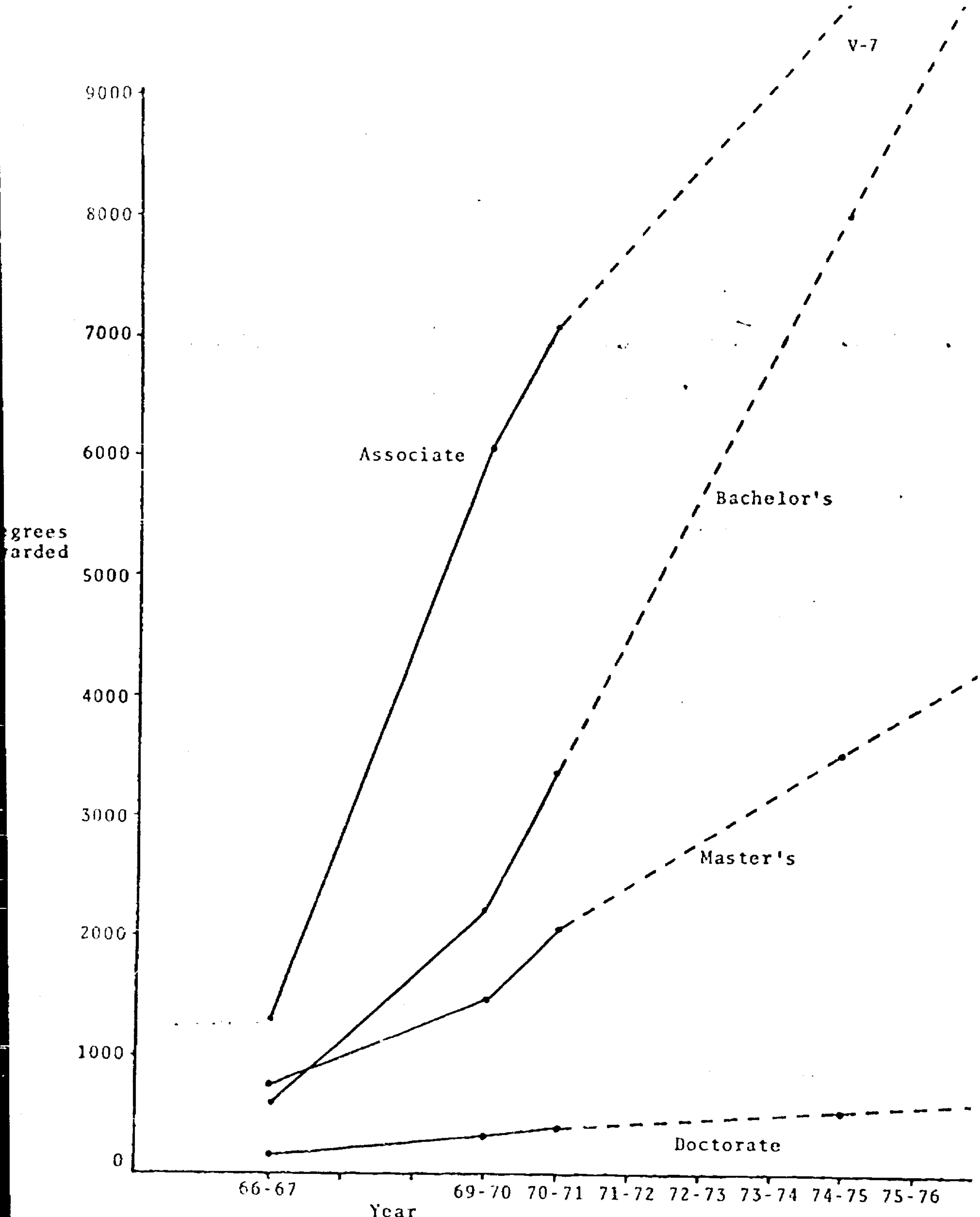
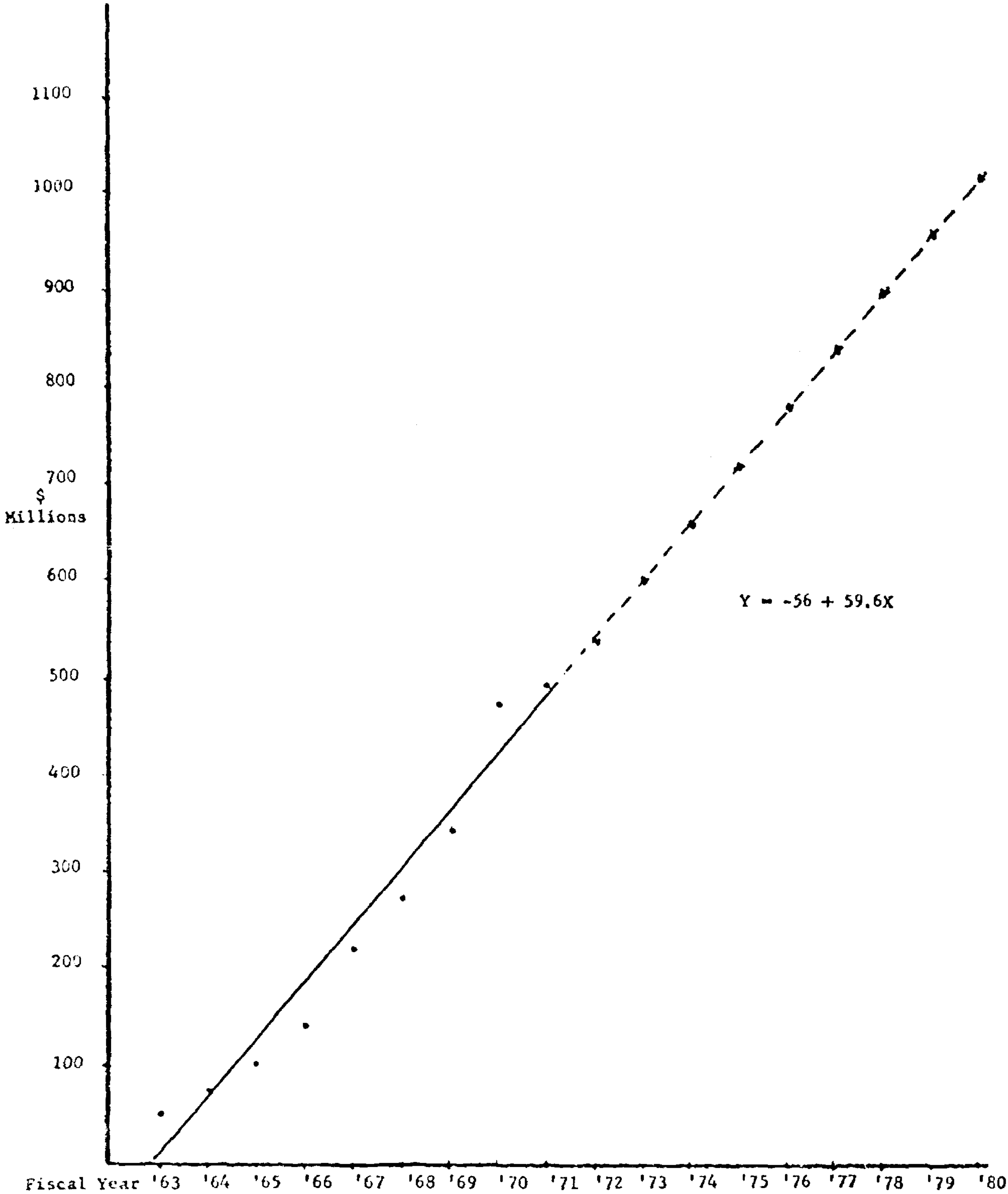


FIGURE V-1
Degrees Awarded in Computer Sciences,
Data Processing, Information Sciences, etc.
(Estimates)

Figure III-12
Estimated Total Expenditures for Computing
in U.S. Higher Education Through FY 1980



Quistorff, Kirk
Results of the Washington High School Computer Programming Survey
Indianola, Washington 98342

This survey was conducted among all of the high schools in the state of Washington during May 1972. There was a 100% return on the questionnaire.

SUMMARY

The following information concerns the teaching of computer programming and the arrangements necessary to provide this activity.

19.8% of the schools offer computer programming and of these 56.5% use a school district computer, 17.1% use a college computer and 17.1% use a classroom computer. Mail or local delivery carry 77.1% of the programs to the computer. Turn around time is one day or less for 51.5% of the schools, 2 to 3 days for 30.3% of the schools and 4 to 5 days for 18.1% of the schools.

The principal languages were FORTRAN and BASIC. 16.8% of the schools teach FORTRAN and 36.4% teach BASIC.

Keypunch machines are used by over half of the schools teaching programming. 26.6% have no keypunch machine while 52.1% of schools have one, and 5.6% have two.

68% of the schools have less than 50 students involved in computer programming. Those programming courses are available to grades 9-12 (22%), grades 10-12 (44%), and grades 11-12 (25%).

The most commonly used computers were manufactured by IBM. 55% of the schools use IBM and the model 360 was the most frequently available.

Two of the earliest courses were started in 1964. During 1969, there was a significant increase in the number of courses.

Schiller, Clark and Gilbert, Bruce
A NATIONAL SURVEY OF THE PUBLIC'S ATTITUDES TOWARD COMPUTERS
Time, Inc.

This survey is concerned with the attitudes of the general public regarding computers. It was conducted by telephone on 1001 adults 18 years of age and older. Respondents were selected at random and represent a cross section of adult population in the United States. Data was gathered during July and August, 1971.

SUMMARY

This survey considers questions of: job involvement with computers, problems people have had with the computer, computers and the consumer, computers in business, computers in government, computers and privacy, computers and the quality of life, and the career opportunities in computer.

Approximately 19% of those surveyed claimed they have had a job requiring direct or indirect contact with a computer.

About 75% of those surveyed reported that they have never had problems in getting a computerized bill corrected. Of those who reported problems, most blamed personnel working with the computer, rather than the computer itself.

89% felt that computers will provide many kinds of information and services to us in our homes; 65% felt computers are helping to raise the standard of living; 68% believe computers have helped increase the quality of products and services.

Areas where the public felt the use of computers should be increased include keeping track of criminals, 78%; guidance of missiles for national defense, 71%; vote counting, 66%, surveillance of activist or radical groups, 56%; projection of election results, 50%; and compiling information files on U.S. citizens, 50%.

58% of those surveyed believe computers represent a real threat to people's privacy versus 54% who disagreed.

Approximately 91% agreed that the uses of computers are affecting the lives of all of us. 75% felt computers will improve our lives and 86% agreed that they will create more leisure time.

Only 12% felt computers can think for themselves, while 23% believed computers of the future might disobey the instructions of people who run them.

Of those surveyed, 76% were in favor of a young person entering the computer field as a career versus 5% who would be opposed.

Walker, Terry
COMPUTER SCIENCE CURRICULA SURVEY
University of Southwestern Louisiana, Spring 1973

This survey was conducted among colleges and universities to determine conditions related to the computer science curriculum. The survey was taken by mail and about 60% of the 490 questionnaires were returned.

SUMMARY

This survey concerned undergraduate, graduate and Ph.D. programs in computer science along with service offerings of departments including available equipment and its use.

Undergraduate

Most undergraduate programs are small (20 or less degrees granted per year) and emphasize training (in order of importance) for: system analyst, graduate programs in C.S., scientific programmer, commercial/systems programmer, D.P. Manager, secondary teacher of C.S., hardware design. In mathematics, the student is expected to take: calculus, linear albrgra, numerical analysis, differential equations, and discrete structures (over 40% of the schools responding indicated these required courses). The undergraduate major is expected to be proficient in (in order of importance) the following languages: FORTRAN, machine or assembly language, COBOL, PL/1, BASIC, ALGOL.

Graduate

Most graduate (master's) programs are very small (73% issued less than 15 degrees in the 1971-72 school year). The objectives (in order of importance) of master's programs were: design software systems, system analyst, prepare for Ph.D. in C.S., scientific programmer, college or J.C. teacher, commercial programmer, hardware systems, D.P. manager, teaching at secondary level. These graduates are expected to possess proficiency in (ordered): FORTRAN, machine or assembly language, ALGOL, PL/1, COBOL/SNOBOL, BASIC/LISP.

Other

The study considers the Ph.D. programs offered at many of the universities. In computer appreciation courses the student was found to run two or more computer programs in 85% of the schools. The most commonly used languages were FORTRAN and BASIC.

The study was concluded by consideration of the available systems and their operation.

CHAPTER XI

COMPUTERS IN RESEARCH

Jim Abrams

Mike Dunlap

David Moursund

Janet Moursund

John Shirey

SECTION A: OVERVIEW OF COMPUTERS IN RESEARCH

"Do cats eat bats?" Alice wondered as she fell down the rabbit hole. Occasionally she asked "do bats eat cats?"; the order didn't seem to matter much, as she didn't know the answer to either question. This is a sort of "cats eat bats" chapter in which we discuss the reversible questions "does research need computers" and "do computers need research." Although the obvious answer in both cases is yes, when one probes for details and tries to see into the future the answers become fuzzy.

DOES RESEARCH NEED COMPUTERS?

And how! There is scarcely an area of research that has not been affected by computers. Statistical analysis of data, and mathematical problem solving are perhaps the most obvious areas in which computers have had an impact. On-line interactive data collection and process control, and computer simulation, have become relatively standard research tools. In this section we will discuss each of these application areas and then make a few comments about the future.

Statistics Suppose one administers a 20 item questionnaire to 50 people. The resulting data has 1000 pieces of information. A few simple statistical computations, such a correlation matrix and an analysis of variance, would require a full day's effort by a skilled desk calcualtor operator. The same computations can be completed in a few seconds on a modern computer. If the questionnaire answers are coded into marksense scan sheets hand labor in the entire process is minimized.

The implication of this example is quite clear: if statistical tools are to be used with any reasonably large data set the computer is a virtual necessity. Computer libraries contain packaged programs which will handle all of the standardly used statistical tests. The researcher needs only to prepare the data in appropriate machine readable form (on punched cards or on marksense scan sheets) and then process the data using a packaged program. To do so requires a little "computer literacy" on the part of the researcher. This "literacy" can probably be acquired in less time than it would take to perform one major statistical analysis using a desk calculator.

It is worth mentioning that few classroom teachers, especially at the secondary school level and below, make use of standard statistics such as mean and variance, or such statistical tools as correlation and regression analysis. A major reason, of course, is that teachers have not had adequate training in statistics and do not appreciate what these tools could do for them. Equally important, however, is the difficulty and time involved in performing the necessary computations on a desk calculator.

Consider the situation of a classroom equipped with a marksense cardreader terminal attached to a computer. Students put their test answers onto marksense cards. In a couple of minutes the test has been graded and recorded, the class mean and variance have been computed, a frequency distribution of scores is printed out and an item analysis has been performed on the test questions. From the point of view of both teachers and researchers the potential for change and improvement of techniques is tremendous. Yet, this is all possible using today's technology; some schools already have such computer facilities, and more will be adding them in the future.

Mathematical Computations The researcher in the sciences often finds himself dealing with large masses of data, hugh algebraic symbol manipulation problems, high order systems of simultaneous functional or differential equations, etc. A computer can aid the researcher in handling many of these problems.

As a simple example, consider the data from an unmanned probe to Mars or Venus. Tens of millions of bits of data are recorded by the instruments onboard the rocket, and sent to receiving stations on Earth. Here the information is stored on magnetic tape to await analysis. Adequate analysis of such masses of data would be impossible without a computer.

An another example, suppose an astronomer is trying to prove the existance of a new planet or comet by a very careful analysis of the orbits of the known planets. The "mathematical" orbits of the known planets can be computed by solving appropriate systems of simultaneous differential equations, but accuracy required in this case can only be achieved by use of a computer.

On-line Systems Join us for a trip through the Chemistry Department at the University of Oregon. We quickly pass by the freshman lab, where students still learn to bend glass, precipitate salts, and handle the basic laboratory techniques, and enter the research labs used by the faculty. In these labs we find hundreds of thousands of dollars of modern equipment: a mass spectrometer, a protein X-ray crystallographic facility, a nuclear magnetic resonance spectrometer, an ultracentrifuge, an infra-red spectrophotometer, etc. And, we find more than a dozen mini-computers! These computers are connected into a mini-computer network which allows them to interface with the university's large computer facility (a PDP-10), with the various major pieces of laboratory equipment, and with each other. Such a computer facility is not extraordinary in modern laboratory research; on the contrary, it is coming to be accepted as the "normal" laboratory setup.

In a typical chemistry experiment a sample to be analyzed is placed into an experimental apparatus which may cost several hundred thousand dollars. This apparatus is then placed under the control of an appropriate program and the experimenter. When the equipment is functioning correctly (the computer can check to see that voltages, magnetic fields, temperatures, vacuums, etc. are at the appropriate levels) the experiment begins. It may last only a fraction of a second, with thousands of pieces of data being fed to the computer memory in one burst of activity. Or the experiment may continue for several minutes, with the researcher monitoring the results and instructing the computer to make changes in temperature, pressure, etc. Data from the experiment passes through the computer memory to magnetic tape, or to the PDP-10. (If massive computations are needed to allow the researcher to modify the conditions of the experiment while it is still going on then the PDP-10 will be used.)

The computer has become a fundamental tool in laboratory research in the sciences. The University of Oregon (a 15,500 student school) has about two dozen mini-computers, located in its various science departments, and is adding more each year. The Computer Science Department regularly offers courses in mini-computers. Similar course offerings are available at Oregon State University and at almost every major university in the United States.

Simulation A major goal of research is prediction--what will happen under a particular set of conditions? When the situation to be predicted is simple (what happens to an object falling under constant acceleration in a vacuum) a simple predictive model suffices ($d=.5at^2$). When the situation is complex an intricate model may be needed. For example, what will be the state of our nation's economy one year hence? Suppose we impose a 10% surcharge on federal taxes and put a price freeze on all foodstuffs? If we had a good simulation model of our economy we could answer such questions accurately and rapidly. Of course, a very fast computer would be needed, and the questions would have to be asked in a form "understandable" to the simulation model.

Computer simulation models have become a standard tool in economics, psychology, sociology, political science, and of course in physical sciences, business, government, and industry. Generally these models are worked on by teams of researchers over a period of years. The models are repeatedly tested and gradually improved. Improvements come from better understanding of the underlying theory, increases in the number of variables considered in the simulations, and greater accuracy in the actual computations (the latter two improvements requiring bigger, faster computers).

The Future By extrapolating current trends we can gain some insight into the future. For example, computerized data banks are growing rapidly; this will surely continue. To name one specific instance, computer-assisted medical checkups are just beginning to be accepted by the public and medical profession. All of the data gathered in such a checkup is in computer readable form. Eventually we will have detailed lifelong medical histories of millions of people. Such a databank will be of immense value to the medical researcher.

On-line data entry, and thus computer analysis, will become common. To site one example, it won't be too long before the grocery checkout stand will consist of a computer terminal and a "wand" which reads a code off of each item purchased. Information on each item sold (including date, time of day, and other items purchased by the same person) will be gathered and subjected to computer analysis. If the purchase is by charge card or check then identification of the purchaser will be done using a computer information retrieval system--and precise buying habits of the person will be subject to computer analysis. Such is the dream of the consumer product analyst!

We noted earlier in this section that simulation models are becoming important in economics, sociology, psychology, etc. This trend will grow. The models will get better and better, and the possibility of a single person being able to create such a model, or even understand fully an existing model, will decrease rapidly. Such research will be the province of well funded teams of specialists.

In the physical sciences we can expect computers to play an ever increasing role. For each subject matter there will eventually be one very large computer center, serving as an interactive information retrieval and problem solving facility. New research results will be deposited in the system. Teams of researchers will develop programs to analyze the data and solve the problems arising in that science. Nationwise time-shared computing networks will be needed for the use of such facilities.

In education we can expect progress towards accountability. Databanks of student and teacher information will grow rapidly, and methods of analysis will improve. We will be able to give meaningful answers to the question of what constitutes good teaching. A careful analysis of the progress of Mrs. Doe's students during the 10 years after they leave her class will give us good insight into Mrs. Doe's teaching skills (unfortunately, 10 years too late). Eventually we will have a good understanding of the relation between things we can readily measure (how well John did on the "standardized tests") and long term implications (John became a successful company president). This will allow us to evaluate teachers, and the other tools of education, much more effectively. It will provide a basis for significant improvement in our educational system.

DO COMPUTERS NEED RESEARCH?

Again, of course, in keeping with the theme of this Handbook this section will focus on research on computer education. While those of us in the field of computer science education are convinced of a growing need for computer literacy and of the appropriateness of computer science courses in the public schools, this conviction is largely based upon faith rather than hard data. Let us explore some of the major "unknowns" and "unprovens" with which computer science education must deal in 1973.

Teaching Computer Programming Computer programming has become a standard course in most colleges and universities, and is now relatively common in the secondary schools. More than a million students per year take introductory programming courses. How do people learn computer programming? Some students find the subject easy and interesting, while others find it dull and difficult. What are appropriate prerequisites for an introductory course? What are suitable goals? How rapidly should a course cover the material, and to what depth? What hardware, software, and textual materials are needed to insure or assist student learning? It is ironic that, given the excellent hardware now available for studying the learning process, the very people who know that hardware best have not applied it to the learning of their own trade.

Even in the introductory computer programming course--the course most often taught today, and the one which has been around the longest--we do not find agreement as to a suitable approach and/or suitable content. For example, one approach is to cover the syntax of a language quite rapidly, and require the students to memorize that syntax. It is found that many students who approach computer programming via such a course do indeed learn to program, and can write programs to carry out the computations needed for the problems in business or science. Many others, however, make little worthwhile progress in such a course.

A second approach is to teach computer oriented problem analysis and to discuss in detail the entire process of problem solving. The syntax of a language is either integrated throughout the course, or is presented in the latter part of the course after the ideas of computer-oriented problem solving are well entrenched. This problem analysis approach to teaching computer programming is receiving considerable attention lately, but there appears to be no research available to support its claims of superiority.

One of the "claims" of many computer programming instructors is that the problem solving skills one acquires in a computer programming course carry over to other fields of study. Similar claims have been made in the past for geometry, Greek and Latin. Evidence to support such claims is lacking; indeed, there is some data which tends to indicate that the reverse is true--that there is no transfer of skills at all. Nevertheless, hope springs eternal... If one could prove that computer programming is a good vehicle for teaching general problem solving skills this would provide a strong justification for the widespread proliferation of such courses.

Computer Literacy In several places in this Handbook the idea of "computer literacy" is discussed and courses designed to teach computer literacy are recommended. The general area of computer literacy is another battleground on which is being fought a series of data-less skirmishes. One school of thought claims that such a course must include computer programming; the other school holds that this is unnecessary--indeed, that it may be undesirable. Until we have good agreement upon what constitutes computer literacy it will be impossible to determine appropriate couse content or to demonstrate that one method of instruction is superior to another. Related questions include: At what age or grade level should computer literacy instruction begin? What teachers (for example, math vs. social science) should teach such courses? We cannot answer such questions without a good understanding of the components of computer literacy.

Vocational Aspects To many people the field of computer science is strictly vocational; one studies computers to qualify for a particular job. What skills are needed for the various professions in the computer field? With detailed answers to this question we could determine the possible role of vocationally oriented computing programs in the secondary schools. (The Sabin Center, in Clackamas County, Oregon, offers such program to secondary school students.) Do the private EDP schools do an adequate job? How do their programs compare with the two year data processing programs of community colleges? What role do the college and university level computer science courses play in the overall vocationally oriented program?

An interesting sidelight concerns the psychology of computer programming. Many professional programmers are secretive and sensitive about their work, and seem unable or unwilling to document their programs adequately. Many work well on one-man projects, but do badly in team project situations. Considerable work has been done on tests to measure the potential success of a student who is thinking of entering the computing field. These tests are reasonable accurate in predicting whether a student will learn to write computer programs, but ignore almost completely the personality characteristics of the successful programmer.

Teaching Computer Science Who should teach computer science courses and what level of training do they need? At the university level the natural tendency and trend is to staff computer science departments by hiring Ph.D.'s in computer science. This will undoubtedly continue, higher education being what it is. But consider the secondary school or small college. The typical secondary school teacher of computer programming has had less than one year (say less than 12 quarter hours) of formal instruction in the field. Moreover, he has had little or no work experience in the field outside of his own classroom experience. Can such a person do an effective instructional job? What sort of training and experience is desirable for the inservice or preservice teacher of computing at the secondary school level?

Hardware and Software Facilities What hardware and software facilities are most appropriate for computer science instruction? A standard controversy has to do with batch processing vs. time-shared computing. Many claim that time-shared computing is far superior as an instructional tool--but good studies on this are lacking. And what of the language to be taught. Many argue the merits of BASIC vs. FORTRAN vs. COBOL, PL/I, etc. Leaders in the computer science field now recommend new languages which allow structured programs to be written, and suggest we should be teaching top-down analysis, step-wise refinement, and structured programming. What languages are most suitable, and what is the role of language in learning how to solve problems? Many difficult but important questions can be raised here.

Teaching Using Computers One chapter of this Handbook is devoted to the area of the computer as an aid to learning, and another chapter is devoted to computer assisted instruction. In computer assisted learning the student makes use of a computer in studying various subjects. He may use a packaged program to solve an equation arising in a math or science course, or he may write his own programs to solve such problems. What are the benefits of computer assisted learning? Does a student learn more, or better, or faster when a computer is available for carrying out the computations which arise in a course? Is it better to have the students write their own programs or use packaged ones? It is claimed that if a student can write a program to solve a particular type of problem then he "really understands" how to solve that type of problem. Is this true? Is time better spent working on a program to solve a problem or in studying the problem area in the traditional way(s)?

Unlike many areas of computers in education, computer assisted instruction has been the subject of considerable research. Unfortunately, much of the research has focused on drill and practice, and even here many questions remain unanswered. Existing literature does not seem to provide answers to such basis questions as the following: What aspects of computerized drill and practice are more conducive to student learning than the standard flash cards or workbook drill? How do the more sophisticated versions of CAI, such as tutorial programs, compare in effectiveness with conventional modes of instruction? What are the actual costs of producing good CAI materials, and what are the necessary qualifications to produce good CAI materials? What are the effects upon students of receiving a significant proportion of their

their instruction in interaction with a machine rather than with people and/or books?

<u>Conclusion</u> To date computers have had relatively little impact upon education below the college level. Even at the college level the impact has not been as widespread as one would expect. Many of the questions listed in this section will need to be answered before the public schools will be willing to commit the funds and personnel needed to allow computers to make a significant impact. Considerable retraining of college and university level faculty needs to occur if students at this level are to gain appropriate insight into the role of computers in research and education.

CHAPTER XII

INSERVICE TRAINING

by

Mike Dunlap

Gerald Larer

Sr. Clare MacIssac

David Moursund

Robert Thomas

Sue Waldman

SECTION A: OVERVIEW

The great majority of all teachers and school administrators have had no formal introduction to computer science and the field of computers in education. Some, of course, have very little need to know anything about this field. However, other sections of this Handbook have presented a convincing and fairly comprehensive overview of the current and potential future role of computers in education. Most educators, both as educators and as "educated" adults, could benefit from an increased knowledge of the computer field.

Each school district should have an overall plan of inservice, workshop, summer program, etc. programs available to their teachers and administrators. Computer science education is one of a large number of subject matter areas that should be included in such a plan. Here the needs of a school district will depend heavily upon the teachers' and administrators' current level of computer literacy, the computing facilities available to the schools, and the computer-related goals of the district. It is not difficult to think of a number of different inservice training opportunities the district might want to make available.

Of very high priority is computer literacy for school administrators and higher level administration. If the district's leaders know nothing about the field then any possible progress in instructional or administrative use of computers will be stifled. Chapter VII Section B of this Handbook discusses workshops for administrators.

The second priority, and the main concern of this section of the Handbook, is computer literacy for large numbers of teachers at all grade levels. One possible solution is to provide good inservice courses carrying college credit.

Inservice training in computing in a school district tends to follow a natural progression over a period of years. The first such course given in a district is usually aimed mainly at math and science teachers. The course is essentially an introductory computer programming course, with some of its examples drawn from secondary school math and science.

A second phase is to offer a course designed for a broader, primarily non-math-oriented audience. This course is perhaps one-half computer programming and one-half computers in education. It should be noted that this second course is much more difficult to teach than the first, straight computer programming course. The instructor must be much more broadly educated in the various areas which make up computer literacy and the field of computers in education.

Additional phases (third and higher) will tend to be courses designed to fit the specific needs of teachers not reached by the first two courses. An elementary school teacher, for example, tends to have relatively little interest in an inservice course that is one-half computer programming in BASIC.

For people who have offered inservice courses such as those described above, and for people who have taken such courses, it is evident that <u>one</u> course generally cannot provide an adequate introduction to the computers in education field. Math and science oriented teachers will need additional coursework in computer programming. Social studies teachers will need considerable additional work in the computer literacy topics--capabilities, limitations, and social implications of computers. Thus a school district needs to consider how its teachers will receive this followup training.

The next section of this Handbook gives a fairly detailed course outline for a "phase 2" inservice course. The course is not particularly math oriented and it is fairly well balanced between computer programming and the overall field of computers in education. The course would also be suitable for use in a college or university. During fall term 1973, for example, the University of Oregon will offer a 3 hour Computer Science 407(g) course which follows this outline fairly closely. (The UO course will probably be split closer to .4 computer programming, .6 overview of computers in education).

SECTION B: A TEACHER TRAINING COURSE

A school's needs for inservice training in the computer field will be highly dependent upon the amount and type of computing facilities available to the school, the level of training already acquired by the teachers, and the school's overall goals in the field of instructional use of computers. The typical inservice course is a compromise. It is designed to serve as wide an audience as possible, thus insuring that the course can be given. It is aimed at teachers from a variety of disciplines, at a variety of grade levels, and from a number of different schools and school districts.

The course described in this section is such a "compromise" course. It is designed to be suitable for both math-oriented teachers and for non-math-oriented teachers. It places considerable emphasis upon computer programming, but still about half of the course is devoted to the more general topic of computers in education.

<u>The goal of the course is computer literacy for teachers.</u> It assumes that computer facilities (at least one remote terminal) will be available at the place where class meetings are held, and that many of the participants will have access to computer facilities in their schools. The outline given here is designed for a 10 meeting course. Each meeting would be about three hours in length. Typically such a course carries three hours of college credit.

The course is based upon the interactive language BASIC. In Oregon approximately half of the secondary schools, representing nearly 3/4 of the secondary school students have access to BASIC.

<u>Helpful Hints for the Instructor:</u>

1. Order films well in advance, especially if the course is to be held during the school year.
2. Order texts and supplementary material well in advance.

3. Develop a personal library of texts and articles for use in the course.
4. Have enough TTY paper and paper tape on hand for each class.
5. Be sure participants understand there will be outside work involved.
6. Before class begins read through the material and make the necessary changes to fit your needs and your computer system, if any.
7. It is assumed that the class discussion portion of the meeting will take one-half to two-thirds of the allotted time, in most class meetings.
8. Establish a time schedule for participants use of TTY.
9. Have a consultant available while participants are using the TTY.
10. Do not assign problems which have for their output a teletype run longer than 3 minutes.
11. Be sure participants with no TTY available for use between class meetings have priority on using TTY during meeting times.
12. The demonstration programs may be used in three ways: on the board, as a handout, and as a prepared program on the TTY.
13. Feel free to make major changes in the course outline to fit your ideas of the proper orientation for the course.
14. When possible, readings should include topics to be discussed at the next class meeting.
15. If books listed for suggested readings are not available, assign similar topics from other sources.
16. BE PREPARED! (Remember Parkinson's Law: "If anything can go wrong, it will; if nothing can go wrong, something will anyhow.")

[illegible] FOR READINGS OF ASSIGNMENT

[illegible]er, Robert; The Digital Villain. Addison Wesley, 1972.

[illegible]all, Marion J.; What is a Computer?. Houghton Mifflin Co., 1972.

[illegible]m, James S., Basic BASIC. Hayden Book Co., 1970.

Computer-Assisted Instruction and the Teaching of Mathematics. National Council of Teachers of Mathematics, Inc., 1969.

[illegible]owley, Thomas; Understanding Computers. McGraw-Hill Inc., 1967.

[illegible], Richard C.; Introduction to Computers and Computer Science. Boyd & Fraser Publishing Co., 1972, San Francisco, California.

[illegible]erald, Curtis F.; Computers and the Art of Computation. Addison Wesley, 1972.

[illegible]mming, Richard; Computers and Society. McGraw-Hill, 1972.

[illegible]nson, M. Clemens; Educational Uses of Computers: An Introduction.

Kemeny, John G.; Man and the Computer. Charles Scribner and Sons; New York, 1972.

Kemeny and Kurtz; Basic Programming. John Wiley and Sons, 1971.

Martin and Norman; The Computerized Society. Prentice-Hall, Inc., 1970; Englewood Cliffs, N. J.

Murrill and Smith; Basic Programming. Intext Educational Publishers; 1971.

My Computer Likes Me*---when i speak in BASIC. Dymax, Menlo Park, Ca. 1972.

Orf and Hestwood; Computer Conversations. The Math Group, 5625 Girard Ave. S., Minneapolis, Minn. 55419, 1973. (These are good Bulletin Board cards.)

Pavlovich and Tahan; Computer Programming in BASIC. Holden-Day Inc., 1971.

Rothman and Mosmann; Computers and Society. Science Research Assoc., 1972.

The School Mathematics Project; Computing in Mathematics: Some Experimental Ideas for Teachers. Cambridge University Press, 1971.

MEETING #1

ACTIVITIES: Introduce and begin the course. Give an overview of computers in the field of education. Introduce the idea of packaged programs and details of how to use them. In particular, give information on how to log on and off the computer system and access a library program.

I Introduction: The instructor should introduce himself and the course. Each "student" should introduce himself. (It is desirable to arrange a car pool.) The introduction should cover the goals of the course, the nature and extent of required readings, homework assignments, term papers, tests, etc.

II What is a computer? Define and describe a computer system. (See Chapter II Section A of this Handbook.)

III Brief overview of computers in education: (see Chapter II Section C of this Handbook.)

IV Films: The following two films are excellent, but overlap considerably. It is recommended that one be shown during the first class meeting and the other one be shown during the last class meeting.

A. The Thinking Machine, Bell Labs, free, 20 minutes (first choice).

B. The Incredible Machine, Bell Labs, free, 20 minutes.

V Packaged programs: Show the class participants how to log on the computer system, run a packaged program, and log off the system. Provide class meeting time for each student to practice this process. Discuss very briefly several short library programs the students can run. All of these should be short and simple. Simple games, little CAI drills, etc. are ideal. It is extremely important that all students learn how to log on, run a packaged program, and log off!!

ASSIGNMENTS:

I. Readings: (These are representative possibilities--don't assign them all!)

A. Chapter II of this Handbook

B. Ball, Marion: What is a Computer? Houghton Mifflin Co.

C. Computers...a Beginning: DCE Books. Unit I on "What is a Computer" is suitable for use here.

D. If the course is using a computer programming text, the first chapter may contain suitable introductory material. Such is the case in Entering Basic by Sacks and Meadows, SRA, 1973.

II. Programming: Each student should be expected to run at least two packaged programs different from those run during the class period; turn in sample output from the program runs and a careful discussion of possible instructional uses of the programs. Orient the discussions to the level at which one teaches.

HANDOUTS: (These will need to be prepared by the instructor, and should fit the particular computer system available to the students.)

I. How to log on, access a packaged program, and log off the system.

II. A list, and brief description, of a dozen or so interesting and appropriately chosen packaged programs available on the system.

III. Sample output from one or two of the programs in II, for class discussion purposes.

IV. Examples of some of the nationally distributed packaged program materials make good handouts. REACT materials (Northwest Regional Labs) and Huntington Project materials (Brooklyn Polytechnic Institute) are well known.

MEETING #2

ACTIVITIES: Introduce at least one new interesting and generally useful packaged program, and discuss the role of packaged programs in instruction. Introduce computer programming in BASIC, and demonstrate that computers can be used to carry out the computation indicated by a formula. Give a very brief introduction to flow charting.

I Review, solidify, and answer questions. Typically about one-half hour should be devoted to this each class meeting. Keep in mind that a week has passed since last class meeting, and many of the class participants will have forgotten much of what was covered in the first meeting. Students who join the course late (i.e., who missed the first meeting) can be eased into the course this way.

II Packaged programs. Get the students to name and discuss several of the packaged programs they found particularly interesting while doing their homework. Suggest, and discuss briefly, several additional packaged programs of general interest. Discuss the topic of possible roles of packaged programs in instruction.

III Introduction to BASIC

A. Sample programs (put on the board, and discuss). Include a very simple flow chart with each program, to begin to introduce the idea of flow charting.

```
10 REM: COMPUTATION OF PAY
20 INPUT H,R
30 LET P = H*R
40 PRINT P
50 END
```

```
10 REM: PERIMETER OF RECTANGLE
20 INPUT L,W
30 LET P = 2*L+2*W
40 PRINT P
50 END
```

B. Key concept: Slight variations on the above programs, and use of +, -, *, /, allow a large proportion of all of the "formula-type" computations in mathematics grades 1-10 to be performed by computer. Note that exponentiation and the BASIC functions such as SQR will be introduced later in the course (or students can learn these topics by reading their text).

C. Entering and running a program. Supervise the students as they enter and run a simple program. It is essential that each student go through this activity and become confident of his ability to do so.

ASSIGNMENTS:

I Reading

A. If the course is using a computer programming text, make an appropriate reading assignment from the text. If no text is being used one will want to prepare and handout material giving the syntax of BASIC along with examples illustrating uses of the various instructions.

B. Chapter IV of this Handbook, which is on the topic of computer assisted learning.

C. What is Timesharing? Computer Science Corp. (A pamphlet).

II Programming

A. Run another packaged program and write up the results as before. If a computer facility is available in your school then select a group of students and demonstrate use of packaged programs to them. Have them run several packaged programs and note their reactions. Write up a brief description of this activity and what you learned from it.

B. Make as long a list as possible of formulas appearing in typical grades 1-10 instruction which can be programmed using the ideas in BASIC covered during this class meeting.

C. Write and run at least two computer programs to carry out computations from formulas compiled for B above. Turn in the entire TTY listing of these computer runs.

HANDOUTS: If a good programming text is not being used it will be necessary to prepare a handout giving details on how to enter a program, correct errors, run the program, etc. The instructor should be aware that some students will encounter problems not covered in the brief introduction to programming so far. Thus, what is an error message, and what does it mean? What happens if the student has line transmission problems, or the computer goes down? A good handout would reassure the student, and tell him what action to take if various things go wrong.

NOTE TO INSTRUCTOR: A major decision needs to be made about whether or not to introduce string variables in the course, and how early to introduce them. Some versions of BASIC do not allow string variables. A widely available version of BASIC (Hewlett-Packard) requires a DIM statement for each string variable. This can be somewhat confusing to students, but certainly doesn't prohibit introduction of string variables early in the course. Dartmouth BASIC (and PDP-10 BASIC, among others) allows string variables to be used without a DIM statement, and then also allow vectors of string variables. In the case that this "nice" BASIC is available it is strongly recommended that string variables be introduced early in the course. Either MEETING #2 or #3 would be an appropriate place to introduce string variables in this particular course outline. Two typical demonstration examples are given below. The first could be used in MEETING #3, and the second in MEETING #4.

```
10  REM:  DEMONSTRATION OF STRING VARIABLES
20  PRINT "HI! I AM A COMPUTER.  WHAT IS YOUR NAME?"
30  INPUT N$
40  PRINT "I AM PLEASED TO MEET YOU ";N$;"."
50  PRINT "WHAT IS YOUR FAVORITE LETTER";
60  INPUT L$
70  PRINT
80  PRINT L$,L$;L$;L$,L$,L$
90  PRINT "I LIKE ";L$; " ALSO ";N$;"."
100 PRINT "BYE FOR NOW."
110 END
```

```
10  REM:  USING STRING VARIABLES INTERACTIVELY
20  PRINT "WHAT IS YOUR NAME";
30  INPUT N$
40  PRINT "ARE YOU MALE (TYPE IN MALE), OR ARE"
50  PRINT "YOU FEMALE (TYPE IN FEMALE";
60  INPUT S$
70  IF S$="MALE" THEN 110
80  IF S$="FEMALE" THEN 130
90  PRINT "I DO NOT UNDERSTAND YOUR RESPONSE."
100 GO TO 40
110 PRINT "I THINK "; N$; " IS A FINE BOY'S NAME."
120 GO TO 140
130 PRINT "I THINK "; N$;" IS A NICE GIRL'S NAME."
140 PRINT "SEE YOU AGAIN SOMETIME ";N$;"."
150 END
```

Notice that neither of the above examples is in any sense "mathematical". The heavy use of such string variable examples is recommended if the course is primarily designed for non-math teachers.

MEETING #3

ACTIVITIES: Review and solifify progress so far. Discuss the use of electronic desk caluclators for "formula" computations. Discuss the overall concept of time-shared computing, with particular emphasis upon the idea of "data banks" (interactive information storage and retrieval) and their implications. Introduce the idea of outputting a character string in BASIC.

I Review, and answer questions. Get the class to suggest formulas appearing in public education grades 1-10 which cannot be programmed using the ideas covered so far. Raise the question of what constitutes a formula. Are the computation of mean, median, mode, or extreme values included under the topic "formula"? Continue to discuss educational implications of using machines to carry out formula computations.

II Electronic desk calculators. Discuss the fact that electronic desk calculators are easy to use and can readily carry out formula computations. An electronic desk calculator is a cheaper, more readily available, and more convenient device for carrying out such computations. This introduces the idea that some problems are better done by hand or desk calculator rather than by computer. If time permits demonstrate one or two different electronic desk calculators.

III Computer Networks. Give an overview of this topic. Some of the major national networks are used to construct and access data banks. An airline, hotel, or motel reservation system would be an example. Stock brokers have terminals in their offices that access data banks of stock information. The US government is a major constructor and user of data banks. A public school is also a major constructor and user, although the data bank may not be computerized.

IV BASIC--printing a character string. Put the following sample programs on the board, and discuss. Include block diagrams and/or brief flow charts.

```
10  REM:  COMPUTATION OF PAY
20  PRINT "ENTER HOURS WORKED AND PAY RATE."
30  INPUT H,R
40  LET P = H*R
50  PRINT "THE PAY IS"; P
60  END
```

```
10  REM:  OUTPUT OF A FIR TREE
20  PRINT "       X"
30  PRINT "      XXX"
40  PRINT "     XXXXX"
50  PRINT "    XXXXXXX"
60  PRINT "       X"
70  PRINT "       X"
80  PRINT "       X"
90  END
```

V Film: Man and the Computer, IBM

ASSIGNMENTS:

I Reading

A. Almost every "computer concepts" or "computer literacy" or "computers in society" text contains a chapter or major discussion on data banks.

B. Literature from OTIS (which maintains a school information data bank) would make a good handout and reading assignment.

C. Ask each student to make a list of as many computerized data banks as he can think of. In particular, they should be able to compile a long list of data banks containing information about themselves.

II. Programming

A. Write (or rewrite from previous programming assignment) at least two formula computation programs to use appropriately printed directions to the user, and labeled output.

B. Write a program to output a pattern, such as a picture, or one's name, etc.

MEETING #4

ACTIVITIES: Introduce the BASIC topics GO TO, IF-THEN, and READ-DATA. Discuss some of the general ideas involved in problem solving using computers. Emphasize the role of flow charts and the concept of "top down" in problem analysis. Note that MEETING #5 will be a continuation of these same activities.

I Brief review. Keep it short this time, as there is a lot of new material to cover in this meeting.

II Looping. Discuss the concept, illustrate using flow charts and sample programs, and emphasize loops as a key concept of programming.

A. Sample program to illustrate a GO TO loop using READ-DATA.

```
10 REM: PAY OF SEVERAL WORKERS
20 REM: ILLUSTRATE GO TO AND READ-DATA
30 READ H,R
40 LET P = H*R
50 PRINT "HOURS=";H,"RATE=";R,"PAY=";P
60 GO TO 30
70 DATA 40,3.75
80 DATA 36,4.19
90 DATA 45,2.87
100 END
```

B. Sample program to illustrate IF-THEN and use of a sentinal value.

```
10 REM: PAY OF WORKERS
20 REM: USE IF-THEN TO TEST SENTINAL VALUE.
30 PRINT "INPUT HOURS WORKED AND PAY RATE."
40 PRINT "ENTER NEGATIVE VALUES TO HALT THE COMPUTATION."
50 INPUT H,R
60 IF H<0 THEN 100
70 LET P =H*R
80 PRINT "HOURS";H,"RATE";R,"PAY=";P
90 GO TO 30
100 END
```

C. Sample program to illustrate a counting loop.

```
10 REM: A COUNTING LOOP USING IF-THEN
20 LET C=1
30 PRINT "I PROMISE TO NOT CHEW GUM IN CLASS."
40 LET C=C+1
50 IF C<=20 THEN 30
60 END
```

III Problem Solving. With the BASIC language tools now available to the student a number of significant and difficult problems can now be attacked. Now is the time in the course to develop the techniques and good habits that will lead to success in attacking new problems. A key concept is that the student will imitate their instructor. Thus in presenting the examples in II above, and in the remainder of the course, use each problem example as an opportunity to illustrate desirable steps and techniques in problem solving. A fairly standard list of steps to follow when solving a problem by computer follows:

1. Understand the problem. Get a clear picture in mind of the nature of the input, the desired process, and the output.
2. Analyze the problem and figure out a solution process. On all but the simplest of problems this will require breaking the problem down into smaller problems (top down analysis).
3. Develop appropriate block diagrams and flow charts. During 2 and 3 one will need to decide upon what data structures are used. (At this point in the course the only structure available is a collection of single numerical variables and/or string variables. Vectors come later in the course. Arrays, linked lists, etc. come in more advanced courses.)
4. Write a program and prepare sample data to be used to test the program. Include adequate REM's for internal documentation.
5. Enter the program, run it, correct the syntax errors, and run it again.
6. Check for program corectness by comparing output with the hand-computed results on the sample data prepared in step 4.

7. Complete the documentation of the program, and write up the overall process (the results of steps 1-6) in a manner acceptable to your course instructor or employer.

ASSIGNMENTS:

I Reading

A. Computers...A Beginning, DCE Books. Unit 2: Computer Problem Solving.

B. George Polya's books and writings on problem solving.

C. Section III.C.5 of this Handbook, on Top Down Analysis.

II Programming: At this point students are ready to begin some significant programming exercises. Initially these should be simple enough to guarentee success (to build student confidence). Each program should be adequately documented and done in a style acceptable to the course instructor. Students shou d be encouraged to seek out or make up their own programming problems. The instructor should provide a list of alternative problems, giving students considerable options. For example, do three of the following. (Note that none involve ar [illegible]ng; that topic will be covered in MEETING #5.)

1. Write a program to output your name 25 times, with the lines numbered 1,2,3,...25.

2. Write a program to output the odd numbers from 1 to 99.

3. Write a program to input 3 numbers and determine if they form a Pythagorean triple.

4. Modify the "pattern" program (MEETING #3, program assignment B) to output 5 copies of the pattern.

5. Write a program [illegible] output a table of the integers from 1 to 25 along with their squares, cubes, and fourth powers.

6. Write a program to output a table of values of corresponding

temperatures for two or three temperature scales (i.e. °F, °C, °K).

7. Suppose a worker is paid "time and a half" for working over 40 hours in a week. Write a program to input hours worked and pay rate, and output total pay in this case.

8. Write a program which inputs two numbers and outputs them in the order larger, smaller.

9. Write a program to input three numbers and output the largest.

HANDOUTS: Many of the programming topics that should be covered in this introductory course are not mentioned and/or covered in detail in this course outline. A good text on introductory programming will fill this void--as will appropriate handouts. For example, students need details on how to LIST, SAVE, UNSAVE, RESEQUENCE, etc. programs.

An appropriate handout for MEETING #4 would be a complete programming assignment properly documented and written up in a style acceptable to the instructor.

NOTE TO INSTRUCTOR: It is common to introduce the FOR-NEXT loop quite early in many textbooks on BASIC. It is evident that it is not an absolutely essential part of the language. Thus many instructors prefer to firmly establish the idea of loops (using GO TO and IF-THEN) before introducing the FOR-NEXT loop. In such a case MEETING #5 would be the first suitable point for discussing FOR-NEXT loops. For purposes of emphasizing that FOR-NEXT is not a critical part of the language we have refrained from using FOR-NEXT examples until MEETING #7. You should feel free to introduce this topic earlier.

MEETING #5

ACTIVITIES: Continue working on problems involving IF-THEN and GO TO loops. Introduce and illustrate loops to sum a list. Discuss and illustrate the topic of computer simulation.

I Review the concepts of IF-THEN and GO TO loops. Note that typically a loop involves initialization, loop body (the compute-part), increment (or otherwise change some variables) and test.

II Summing. Present and discuss several sample programs similar to the following.

```
10  REM:  TOTAL PAYROLL
20  LET T=0
30  PRINT "ENTER HOURS AND RATE.USE NEGATIVE HOURS TO STOP."
40  INPUT H,R
50  IF H<0  THEN 100
60  LET P = H*R
70  LET T = T+P
80  PRINT "HOURS=";H,"RATE=";R,"PAY=";P
90  GO TO 30
100 PRINT "TOTAL PAY FOR ALL WORKERS IS";T
110 END
```

```
10  REM:  MEAN OF N SCORES
20  PRINT "INPUT THE NUMBER OF SCORES"
30  INPUT N
40  LET T=0
50  LET C=1
60  PRINT "INPUT SCORE NUMBER";C
70  INPUT S
80  LET T=T+S
90  LET C=C+1
100 IF C<=N THEN 60
110 LET M=T/N
120 PRINT "THE MEAN IS";M
130 END
```

Remember that each example such as those above should be presented as a problem analyzed, flow charted, and then programmed. A program such as the one to compute means contains flaws that can be discussed. For example what happens if the value of N which is inputted happens to be = zero? What happens if the user of the program types in a wrong score and realizes it just after doing a carriage return? The above program can also be modified so that it counts the number of scores, detecting the end of the data via test for a sentinal value.

III. Computer simulation. Simulation is one of the fundamental tools of science and social science. Most "real life" sized simulations are done using a computer to carry out the necessary computation. It is certainly worthwhile to spend at least half of one class meeting (i.e., 1 1/2 hours) on this topic. Part of the time, perhaps a half hour, should be spent on a general overview. The remaining time should be spent studying and running one or two instruction-oriented simulations. The Huntington II simulations such as BUFLO, POLUT, POLICY, STERL2, etc. are all excellent for this purpose. The major point here is to devote sufficient time to the activity so that a simulation can be run several times under a variety of conditions. If time and computing facilities permit divide the class into teams and have them compete. For example, see which team can do the best job of wiping out theflies in STERL2. Make sure all class members participate, and learn how to use these programs.

ASSIGNMENTS

I. Reading

A. Chapter IV Section B of this Handbook is on simulation.

B. Most computer science (as distinguished from computer programming) introductory texts contain a chapter on simulation.

C. Be sure to keep making reading assignments in the programming text, if one is being used for this course.

D. Documentation on computer simulations, such as the Huntington II materials, is useful reading.

II Programming

A. Give the students a list of a dozen or so problems involving summing of lists and related ideas covered so far in the course. Ask them to do one or two of them.

B. Each student should be expected to study and understand one simulation other than the one(s) used for demonstration purposes in class. The assignment should be to discuss and run the simulation, giving sample output and noting possible uses in the class. If a terminal is available to the teacher in his school, the assignment should be to try out a simulation in his class, or teach a fellow teacher to use a simulation program.

HANDOUT: Handout a list of simulation programs available in the computer system library. The list should include a brief discussion of each simulation and suggestions for use.

MEETING #6

ACTIVITIES: Spend some time discussing the various simulations which were available for the class to use during the past week, and their experiences in using them in their schools. Work through several programming problems of the level of difficulty of those assigned to the class. Discuss TAB and use of paper tape if they haven't been illustrated in previous classes. Introduce and illustrate RND, the random number generator.

I Simulations: The instructional use of computerized simulations is one of the more important parts of this course. Make sure all class participants have been successful in running one or two simulation programs.

II Programming: Avoid the temptation to rush ahead. Instead, go through in detail possible solutions to several of the previously assigned problems. Point out good and bad features of possible solutions. Indicate typical difficulties apt to be encountered in solving the problems. A good aproach here is to have some of the students put their solutions on the board and then have you and the class discuss the solutions.

III New programming topics

A. TAB (Handing out a sample program illustrating a possible use will probably be adequate.)

B. Paper tape. If the terminals being used are connected to a computer via long distance phone lines, or the computer is only "up" a limited number of hours per day, students will want to learn to use paper tape. Paper tape is also useful to contain a set of data which the teacher would like each student to process.

C. RND Most versions of BASIC allow one to generate a repeatable sequence of random numbers and also to generate a non-repeatable sequence (i.e., one starting in a random place). This aspect of RND is best omitted from the course at this point, as it tends to confuse the student.

In the examples below, we use RND (note no argument), although some BASIC systems require an argument also be included, such as RND(1).

Notice that the second example uses INT. Such topics are probably best introduced in this type of context--i.e., in a useful situation.

```
10  REM:  SIMULATION OF TOSSING A COIN
20  REM:  PERFORMS 100 TOSSES AND OUTPUTS RESULTS
30  REM:  N IS COUNTER OF NUMBER OF TOSSES
40  REM:  H IS COUNT OF NUMBER OF HEADS
50  LET H=0
60  LET N=1
70  IF RND>=.5 THEN 110
80  PRINT "H";
90  LET H=H+1
100 GO TO 120
110 PRINT "T";
120 LET N=N+1
130 IF N<=100 THEN 70
140 PRINT
150 PRINT "TOTAL HEADS IS";H
160 PRINT "TOTAL TAILS IS";100-H
170 END
```

```
10  REM:  SIMULATION OF TOSSING A DIE
20  REM:  OUTPUTS RESULTS OF EACH OF 100 TOSSES
30  REM:  N IS COUNTER OF NUMBER OF TOSSES
40  LET N = 1
50  LET T = 1 + INT(6*RND)
60  PRINT T;
70  LET N = N+1
80  IF N< = 100 THEN 50
90  END
```

ASSIGNMENTS:

I Reading: If a programming text is being used then assign the appropriate sections.

II Programming:

1. Write a program to output 20 copies of your name; the first copy should start in column 1 of the paper, the second copy in column 2, the third copy in column 3, etc. Use the TAB function and a loop.

2. Modify the die tossing example presented in class so that it also counts and outputs the total number of 1's, 2's, 3's, 4's, 5's, 6's, which occur.

HANDOUTS: Prepare and give out a detailed handout on how to use the punched paper tape facility. If the use of punched paper tape is important to the proper use of the computer system available in the schools, make an assignment involving its use.

NOTE TO INSTRUCTOR At this stage of the course some of the students will begin to fall seriously behind in doing programming assignments. They will find that programming is hard and/or that they don't have the knack for it. Part of the difficulty will be failure to spend adequate time preparing a program before going to the terminal. In any case from this point on one should provide considerable programming assignment options. The student who is having trouble in programming should be able to pass the course (indeed, feel successful in the course) by dint of running a larger number of programs, more closely imitating programs used in class lectures, doing reading on a special topic of interest, etc.

MEETING #7

<u>ACTIVITIES</u>: Introduce and discuss the FOR-NEXT loop (if you have managed to avoid it so far--most likely it will have slipped in earlier since students can read books on BASIC). Discuss the topic of table construction. It is a good review topic. The major topic for this meeting is computer assisted instruction.

I Table construction. One of the more useful, and easy, applications of computers is to construct a table of values of a function. For example, the function $F = \frac{9C}{5} + 32$ relates degrees centigrade to degrees Fahrenheit. A table of values is generated by the following program.

```
10 REM:  TABLE OF DEGREES CENTIGRADE AND FAHRENHEIT
20 PRINT "CENTIGRADE","FAHRENHEIT"
30 FOR C=0 TO 100
40 LET F = 32 + (9*C)/5
50 PRINT C,F
60 NEXT C
70 END
```

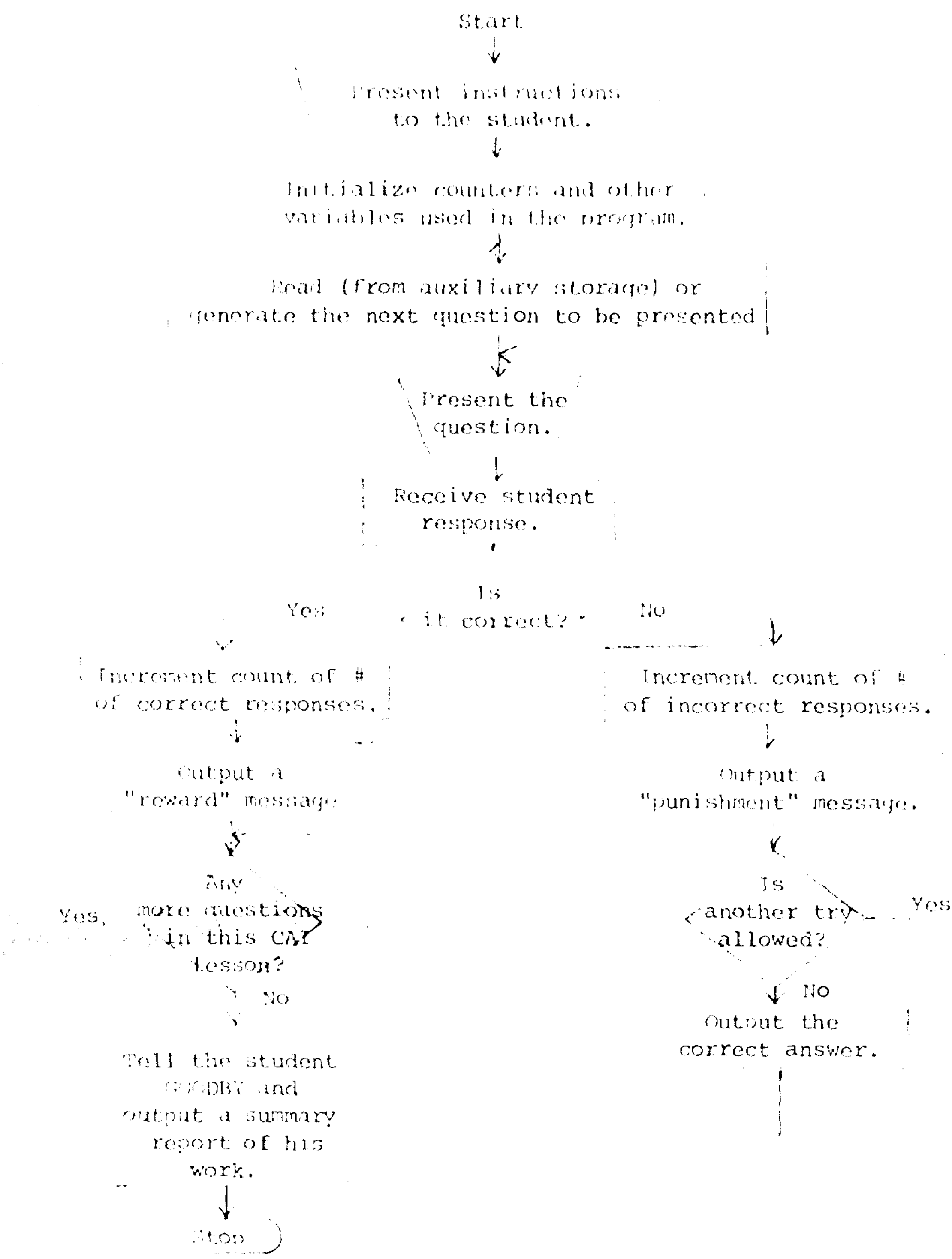
Similarly, one can print tables of square roots, cube roots, etc.

```
10 REM:  TABLE OF SQUARE, CUBE, AND FOURTH ROOTS
20 FOR N=1 to 10 STEP .1
30 LET A=SQR(N)
40 LET B=N ↑ (1/3)
50 LET C=N ↑ (1/4)
60 PRINT N,A,B,C
70 NEXT N
80 END
```

II CAI Sample programs: The following two programs illustrate two simple-minded approaches to writing CAI programs. Present and discuss them. Note that each presents directions to the user, and then a sequence of questions. For each question the answer is received and checked for accuracy, and an appropriate "reward" or "punishment" is given. Records of total correct responses are kept. At the end of the program a report of the total results is printed out. These are all features of almost all CAI drill programs.

```
10  REM: CAI ADDITION DRILL (5 QUESTIONS)
20  PRINT "TYPE IN THE CORRECT ANSWERS."
30  PRINT "PUSH THE RETURN KEY AFTER EACH ANSWER."
40  LET C=0
50  FOR N=1 TO 5
60  READ A,B
70  PRINT "WHAT IS"; A; " +";B;
80  INPUT R
90  IF R = A+B THEN 130
100 PRINT "SORRY, THAT IS INCORRECT"
110 PRINT "THE CORRECT ANSWER IS"; A+B
120 GO TO 150
130 LET C =C+1
140 PRINT "GOOD, THAT IS CORRECT"
150 NEXT N
160 PRINT "THAT COMPLETES THIS DRILL"
170 PRINT "A SUMMARY OF YOUR RESULTS FOLLOWS."
180 PRINT "CORRECT RESPONSES=";C
190 PRINT "INCORRECT RESPONSES=";5-C
200 DATA 15,27
210 DATA -5,18
220 DATA 17,9
230 DATA -8,25
240 DATA 2,9
250 END
```

```
10  REM:  CAL MULTIPLICATION DRILL USING RND
20  PRINT "TYPE IN THE CORRECT ANSWER AND THEN"
30  PRINT "PUSH THE -RETURN- KEY."
40  LET C=0
50  FOR N=1 TO 10
60  LET A=-5+INT(10*RND)
70  LET B=1+INT(9*RND)
80  PRINT WHAT IS";A;" TIMES";B;
90  INPUT R
100 IF R=A*B THEN 140
110 PRINT "NO,THAT IS NOT CORRECT"
120 PRINT "THE CORRECT ANSWER IS"; A*B
130 GO TO 160
140 PRINT "GOOD"
150 LET C=C+1
160 NEXT N
170 PRINT "THAT COMPLETES THIS DRILL"
180 PRINT "YOU GOT";C;" CORRECT, OUT OF 10 QUESTIONS."
190 END
```



Flow chart for a general CAI lesson.

III CAI--general discussion. Present an overview of the current status of CAI, and its current impact upon the world of education. Give special attention to the PLATO project at the University of Illinois, and its very nice student terminal.

IV CAI--packaged programs. If your computer system library contains some worthwhile CAI materials, discuss a course or lesson in class, and allow the class to run it. The Pat Suppes drill and practice in arithmetic grades 1-6 CAI materials would be excellent for this purpose.

ASSIGNMENTS:

I Reading: Chapter V, The Computer As Teacher, in this Handbook gives a good introduction to CAI and related topics.

II Programming

A. Write a CAI drill and practice program on a topic of interest to you. Follow the general ideas illustrated in the flow chart on CAI.

B. If a computer terminal is available in your school, try out some CAI materials (from the computer library, or those written in A above) on several students.

HANDOUTS: When you begin to discuss longer programs or flow charts in class it is highly desirable to prepare handouts of them for the class. Then use an overhead projector, or write the material on the board before class or while students are working on other things. Don't waste class time in a massive "copy off the board" activity!

NOTE TO INSTRUCTOR If you want to assign a major "term project" programming assignment, CAI provides an excellent opportunity. All students should be able to produce some sort of CAI material (perhaps by very close imitation of the examples presented in class). Students who have a knack for programming can produce interesting and worthwhile CAI lessons.

Notice here also that if you have introduced string variables then the CAI lessons can be more interesting, and allow words or other character strings to be typed in as answers by the lesson user.

MEETING #8

ACTIVITIES: This meeting should be devoted primarily to the BASIC topic "vectors" (also known as one dimensional arrays, or lists). Allow a significant proportion of the class meeting time to individual help of students working on programming problems.

I Review. Select an appropriate example and discuss it. A simple game playing program might be a fun thing to present here. The 15 pennies game would be appropriate. In this game one starts with 15 pennies; the person and the computer alternately take turns selecting 1, 2, or 3 pennies. The object of the game is to force your opponent to take the last penny. An example like this can be discussed by giving the students a flow chart and a program, and then tracing through the major parts of the program. A sample solution is given below:

```
10  PRINT "THIS GAME INVOLVES 15 PENNIES. WHEN IT IS YOUR TURN"
20  PRINT "YOU MAY TAKE 1, OR 2, OR 3 PENNIES.  THE OBJECT OF"
30  PRINT "THE GAME IS TO FORCE YOUR OPPONENT (IN THIS CASE."
40  PRINT "THE COMPUTER ) TO TAKE THE LAST PENNY."
50  PRINT "WOULD YOU LIKE TO PLAY A GAME (YES,NO)";
60  INPUT R$
70  LET N=15
80  IF R$="NO" GO TO 45Ø
90  PRINT "WOULD YOU LIKE TO GO FIRST (YES,NO)";
100 INPUT R$
110 IF R$="NO" GO TO 24Ø
120 PRINT "GOOD. YOU MAY GO FIRST."
130 PRINT "INPUT A 1,2,OR 3"
140 INPUT M
150 IF M <> INT(M) GO TO 37Ø
160 IF M<1 GO TO 39Ø
170 IF M>3 GO TO 39Ø
180 IF M>N GO TO 41Ø
190 LET N=N-M
200 PRINT "THAT LEAVES ";N;" PENNIES ON THE BOARD."
210 IF N=Ø GO TO 32Ø
220 PRINT "NOW IT IS THE COMPUTER'S TURN."
230 IF N<=4 GO TO 29Ø
240 LET M=1+INT(3*RND)
250 PRINT "THE COMPUTER TAKES ";M;" PENNIES."
260 LET N=N-M
270 PRINT "THAT LEAVES ";N;" PENNIES.  NOW IT'S YOUR TURN."
280 GO TO 13Ø
290 LET M=N-1
300 IF M=Ø GO TO 34Ø
```

```
310 GO TO 25Ø
320 PRINT "THAT ENDS THE GAME. THE COMPUTER WINS."
330 GO TO 5Ø
340 PRINT "THE COMPUTER MUST TAKE THE LAST PENNY. THUS YOU"
350 PRINT "HAVE BEATEN THE COMPUTER!!!!!"
360 GO TO 5Ø
370 PRINT "THE MOVE MUST BE AN INTEGER..."
380 GO TO 13Ø
390 PRINT "THAT IS NOT A LEGAL MOVE!!!"
400 GO TO 13Ø
410 PRINT "THERE ARE NOT THAT MANY PENNIES LEFT. YOUR"
420 PRINT "MOVE THIS TURN CANNOT EXCEED ";N;" PENNIES."
430 PRINT "WHAT NUMBER DO YOU WANT";
440 GO TO 14Ø
450 END
```

11 Vectors in BASIC. This is an excellent topic to introduce by means of an example that is difficult or impossible to do without vectors. Such a problem follows:

> Write a program to input N and then input N student body numbers and the corresponding scores made by these students on a test. Compute and output the class mean. Then for each student output the student number and the difference between the student's score and the class mean.

First present the problem and work out a solution for the special case N=3, not using vectors.

```
10  REM:  DEVIATION FROM THE MEAN
20  PRINT "INPUT A SBN AND TEST SCORE."
30  INPUT N1,S1
40  PRINT "INPUT A SBN AND TEST SCORE."
50  INPUT N2,S2
60  PRINT "INPUT  A SBN AND TEST SCORE."
70  INPUT N3,S3
80  LET A = (S1+S2+S3)/3
90  PRINT "THE CLASS MEAN IS";A
100 PRINT "SBN","DEVIATION"
110 PRINT N1,S1-A
120 PRINT N2,S2-A
130 PRINT N3,S3-A
140 END
```

The key point is that the length of this solution is proportional to the number of students in the class. Imagine the typing effort if there were 30 students in the class. Also note that using this "model" requires the program to be written specifically for a particular N. Next contrast the above with a solution using vectors.

```
10  REM:  DEVIATION FROM THE MEAN
20  DIM N(100), S(100)
30  REM:  ASSUME NO MORE THAN 100 STUDENTS
40  PRINT "HOW MANY STUDENTS IN THE CLASS";
50  INPUT T
60  FOR K=1 TO T
70  PRINT "INPUT A SBN AND TEST SCORE."
80  INPUT N(K),S(K)
90  NEXT K
100 REM:  HERE WE SUM THE SCORES
110 LET A=0
120 FOR K=1 TO T
130 LET A=A+S(K)
140 NEXT K
150 REM:  COMPUTE THE CLASS MEAN
160 LET A=A/T
170 PRINT "THE CLASS MEAN IS";A
180 REM: COMPUTE AND OUTPUT DEVIATIONS
190 PRINT "SBN","DEVIATION"
200 FOR K=1 TO T
210 PRINT N(K),S(K)-A
220 NEXT K
230 END
```

The above program is representative of the statistical programs that can be readily written in BASIC and/or which one will find in a computer system library. The use of computers to carry on statistical computations is one of the major uses of computers. It would be appropriate at this point to provide a handout on some of the library statistical programs.

ASSIGNMENTS

I Reading: If a programming text is being used students should be directed to the section on vectors.

II Programming: These exercises are designed to provide familiarization with the ideas of vectors. Assign a couple of problems of a degree of difficulty similar to the following.

1. Write a program to INPUT 10 numbers into a vector and then print the numbers out 5 per line. (Hint: use of the "print-comma" option will produce 5 per line output).

2. Write a program to READ 12 numbers into a vector, and then output them in the reverse of their original order (i.e., output the 12th, the 11th, the 10th, etc.)

3. Use the RND function to put 50 random numbers into a vector. Then output all of the values that are ≥ .5; then output all values < .5.

MEETING #9

ACTIVITIES: Continue working on use of vectors. Illustrate several major applications of vectors. Spend some time reviewing the course.

I Indexing into a vector. Problems involving doing frequency counts for several or a large number of categories can often be done simply using vectors. The following example illustrates the point.

```
10  REM:  FREQUENCY COUNT IN DIE TOSSING
20  REM:  SIMULATES 100 THROWS OF A DIE
30  DIM F(6)
40  FOR K=1 TO 6
50  LET F(K)=0
60  NEXT K
70  FOR K=1 TO 100
80  LET T= 1+INT(6*RND)
90  LET F(T)  = F(T)+1
100 NEXT K
110 FOR K=1 TO 6
120 PRINT "THE NUMBER OF ";K;"'S WAS";F(K)
130 NEXT K
140 END
```

II Extreme values in a vector. Finding the maximum or minimum in a list is good practice for the more important problem of ordering a list.

III Ordering a list. This is one of the "classical" problems of most introductory programming courses. It makes a good final example to be done in class by the instructor. All students can understand how to do a sort (say order from largest to smallest) by hand. But translating this into a computer program is a major task. It involves careful problem analysis; top down analysis and problem segmentation are well illustrated in this problem.

The general sorting problem also provides a place to discuss briefly what computer scientists do. There are entire books on searching and

and sorting. The techniques illustrated in a single sorting algorithm (such as the one you will discuss in class) are probably not suited to handling a list of 15,000 student names, addresses and test score data (as one might find in a university, or school district). Be sure to point out that ordering lists of numbers and alphabetizing a list of names are nearly the same problem.

IV Review. Make an outline or list of the highlights of the course. If a final exam is to be given make it clear what will be covered in the exam.

ASSIGNMENT: Don't pile on additional reading and programming assignments at this point. Perhaps assign students to run a packaged program for sorting a list, or for carrying out a statistical computation.

MEETING #10

ACTIVITIES: In a typical course the final class meeting will be devoted to tying up loose ends, answering last minute questions, listing and discussing a few topics the course did not cover, discussing possibilities for further training, evaluation of the course, evaluation of the course instructor, and evaluation of the students (a final exam?).

Section C: Abstracts of related articles for Chapter XII

American Federation of Information Processing Societies, Inc.
"Computer Education for Teachers"
210 Summit Ave., Montvale, N. J. 07645
August 1970

This guide is intended for those people concerned with planning computer courses for teacher training; at the same time, it can readily be used for planning computer courses for a high school curriculum.

All students need to understand the nature and the uses of computers in modern society. For this reason, it is essential that teachers of all subjects have a knowledge of computers. Such instruction should be included in the initial training of teachers and should also be available to teachers in service.

The following content should be included in an introductory computing course:

Information processing (history and need) and the nature of the digital computer as an automatic information processor (taking in, processing, giving out information);

The analysis and organization of information and what contributions the computer can make;

The general characteristics of a programming language, the need for languages that relate to the machine on a one-one basis (assembly and machine languages), and the need for languages that facilitate the communication of the problem itself to the computer (e.g. BASIC, FORTRAN);

The influence of the computer in education, and its effects on society;

Some successful methods for teaching basic computer concepts to the students.

In summary, this article can be useful both to the teacher designing a beginning computer science course for teachers or for students, and to the teacher already instructing a computer science course.

Jay Dee Smith

Atchison, William F.
The Impact of Computer Science Education on the Curriculum
Mathematics Teacher Vol. 66 No. 1, January 1973
Pages 7; 81-83

Computers have become an important force in our society and the educational process should begin reflecting this in teaching about computers, their use, and the social role they play. The most logical person to carry computer science education into the secondary school is the mathematics teacher since he can readily learn the methods of computer science and use them as a tool in his teaching.

At present a teacher training problem exists in our colleges. The World Conference on Computer Education in 1970 called on educators to provide an early introduction to computer science as an integral part of general education in primary and secondary schools and to attempt to use computer aided activities in the classroom when appropriate. A Working Group on Secondary Education was set up by the International Federation for Information Processing and they quickly recognized that the biggest problem associated with the use of computers in secondary education was that of teacher training. Consequently, in 1970 they published a booklet "Computer Education in Secondary Schools--An Outline Guide for Teachers" which has since been revised. They are now in the process of developing a series of ten booklets to supplement this guide, which will then become the content of a college course for secondary teachers faced with the problem of computer education in their teaching. It is hoped that this series will call attention to the need for computer-sophisticated teachers and that as a result many colleges will establish courses with the express purpose of preparing high school and junior high school teachers to teach computer science.

We cannot afford to teach computer science poorly. We must learn from past experience: when the "new" math was improperly taught it alienated many from math for life.

Some colleges are doing something about the problem; and in fact it is possible to obtain a masters degree in the teaching of computer science in a few places. But not enough is being done. It is hoped that more colleges will institute courses designed for teachers at three levels: 1) those teachers who will teach computer appreciation or computer literacy courses; 2)those teachers who will use computer methods in their courses; and 3) those who will teach computer science courses at the high school level.

Don Gallagher

Spencer, Donald
A Guide To Teaching About Computers in Secondary Schools
Abacus Computer Corporation, 1973; 137 p, $12.95

Donald Spencer has been actively engaged in the computer science education field for over 15 years. He is the author of a number of books in the field. In this book he presents many of the fundamental ideas involved in the instructional use of computers at the secondary school level. The book contains three chapters on Computer Science in the Secondary School Curriculum and six chapters related to Methods of Teaching Computer Science. Most chapters are relatively short, and their content is limited. A chapter on films and a chapter on secondary school textbooks are particularly useful. Sources of free and inexpensive materials are given. A number of the chapters contain information that is already evident to almost any teacher who has been involved in the computers in education field. In this regard the book is somewhat disappointing, as there is considerable need for greater "content" material on the topics under consideration. This book is also somewhat disappointing in terms of its quite high price/page ratio.

David Moursund

CHAPTER XIII

SOURCES OF ADDITIONAL INFORMATION

by

Ron Boys

Fred Daniels

Dave Dubose

Mike Dunlap*

Marlyn Kern

*Mike Dunlap, Department of Computer Science, University of Oregon, served as Editor for this chapter.

SECTION A: BOOKS

by Ron Boys

Introduction

The intent of this section was to choose a limited number of textbooks and write reviews to give some idea of their suitability for use in the high school classroom. No attempt was made to give an extensive bibliography of computer science books.

A comprehensive bibliography of computer science books is available from:

The National Council of Teachers of Mathematics
1201 Sixteenth Street N.W.
Washington, D.C. 20036

Also The Directory of Data Processing Education, second edition, 1972-73, is a good guide to computer science books in print. It is available from:

Data Processing Horizons
P.O. Box 4123
Diamond Bar, California 91765

"The purpose of the directory is to provide the data professional with the wherewithal to keep abreast by continuing and broadening his education and knowledge; to aid those responsible for the education of new individuals into the profession in knowing of the materials available to assist them in this endeavor."

Another source of computer science books is Computer Instruction: Planning and Practice, Judith B. Edwards, 1969, available from:

Northwest Regional Educational Laboratory
400 Lindsay Building
710 S.W. Second Ave.
Portland, Oregon 97204

REVIEWS OF SELECTED BOOKS

Albrecht, Robert L., Lindberg, Eric, and Mara, Walter
Computer Methods in Mathematics, Addison-Wesley Publishing Co., 1969, 204 pp., $4.95.

This book is designed for people who wish to learn how to use computers as problem-solving tools. It may also be used as a primary text for a course in computer-assisted problem solving or computer-oriented mathematics. This text could also be employed as a supplementary text for mathematics courses in which the computer is used to demonstrate concepts, or where the computer is used as a laboratory for experimentation, discovery, and the solving of mathematical problems.

The many examples are complete with flowcharts and actual solutions obtained by the computer. Exercises are graded into three levels of difficulty and are presented in special fashion, each building upon the previous. It seems that the book was designed for the high school student who has had at least one year of algebra.

Davidson, Charles H. and Koenig, Eldo C.
Computers: Introduction to Computers and Applied Computing Concepts, John Wiley and Sons, 1967, 596 pp., $12.50.

An intensive survey of computers requiring only high school mathematics. Here is a clear explanation of what computers are, their uses, and the extent of their social and economic implications.

This text was written for college students but it could serve as an introduction to computer science for selected high school students, with ample freedom for the teacher to select those aspects of the subject he wishes to emphasize.

FORTRAN is the principal language used and is introduced in the second chapter. The material was developed to give a student a working knowledge of programming and illustrates underlying concepts without mathematical rigor.

Forsythe, Alexandra I., Keenan, Thomas A., Organick, Elliot I., and Stenberg, Warren
Computer Science: A First Course, John Wiley and Sons, Inc., 1969, 553 pp., $10.50.

Here is a textbook that is definitely geared to the high school student. The book is an outgrowth of a School Mathematics Study Group (SMSG) program begun in 1964. Much of the material in this edition was contributed by committees involving many people.

The thirteen chapters that make up this book are divided into three parts. Part I or the first five chapters is a basic introductory unit. Chapters six through eight mkae up part II and deals with numerical applications. Part III is devoted to nonnumerical applications of computing, symbol manipulation, representative of the more recent areas of computer science research. In its entirety the thirteen chapters of this book constitute a challenging first course in computer science for students with a good mathematical background and plenty of time and energy. Part I alone would provide a challenging one semester course for the average high school student.

Companion computer language supplements such as BASIC, FORTRAN, PL/I, COBOL, and ALGOL provide the specific syntactic details of computer language. The flow-chart language used in this textbook deals only with concepts of central interest to all programming languages.

A big advantage for a beginning teacher of computer science is the availability of a detailed teachers commentary. Part one of each teacher's commentary chapter parallels the student text section by section. Each section begins with a short summary of the corresponding flowchart text followed by general comments and information. Problems and exercises of special importance are emphasized. Detailed remarks and amplification of solutions to student exercises are found here. Supplementary exercises are also found in the teacher's commentary.

Much thought has gone into the writing of this text and the authors seem to be able to anticipate the problems a teacher will have in teaching the course. Although this book is four years old it would seem to be a good choice for a computer science textbook.

Gear, C. W.

Introduction to Computer Science, Science Research Associates, Inc., 1973.

In the introduction the author says, "the objective of this book is to introduce you to the basic principles of computer usage." Whether the student plans to major in computer science or is taking it as a terminal course this introductory material is appropriate.

Companion computer language supplements such as BASIC, FORTRAN, PL/I, COBOL, and ALGOL provide the specific syntactic details of computer language. The flow chart language used in this textbook deals only with concepts of central interest to all programming languages.

A big advantage for a beginning teacher of computer science is the availability of a detailed teachers commentary. Part one of each teacher's commentary chapter parallels the student text section by section. Each section begins with a short summary of the corresponding flowchart text followed by general comments and information. Problems and exercises of special importance are emphasized. Detailed remarks and amplification of solutions to student exercises are found here. Supplementary exercises are also found in the teacher's commentary.

Much thought has gone into the writing of this text and the authors seem to be able to anticipate the problems a teacher will have in teaching the course. Although this book is four years old it would seem to be a good choice for a computer science textbook.

Gear, C. W.

Introduction to Computer Science, Science Research Associates, Inc., 1973.

In the introduction the author says, "the objective of this book is to introduce you to the basic principles of computer usage." Whether the student plans to major in computer science or is taking it as a terminal course this introductory material is appropriate.

This computer science text is intended to be used with companion language manuals such as BASIC, FORTRAN, PL/I, ALGOL, or WATFOR. The author assumes no specific prerequisites in math or computer science because the material is for a first course. Although without two or three years of high school mathematics one would have to skip chapter nine on numerical methods or the student would be lost.

The text covers computer organization, flowchart language, computing systems, errors, data structures, non-numerical applications and numerical methods. This is a carefully planned book that would be appropriate for use in a high school computer science course provided the student had a fairly strong background in mathematics.

Ralston, Anthony

Introduction to Programming and Computer Science, McGraw-Hill Book Co., New York, 1971, 513 pp., $9.95. (From Computing Reviews 1971)

"One result of any college education should be an ability to study and learn the prosaic details of a subject on one's own; much of the study of any computer language falls in this domain." This is quoted from the preface to the book; the preface alone is a significant contribution to computing literature.

How refreshing it is to pick up an introductory text in which the author faces squarely the many problems involved in teaching computing, states clearly the choices he has made and then backs up his choices throughout the book. This is not a collection of classroom notes; this is a carefully planned book.

one can disagree with the choices that the author has made, but one is forced to respect anyone who knows exactly where he is going.

This book does not use the ANSI standards for flowcharting, and there is the common passion for having all decisions binary in their outcomes. The author has an optimistic view of a beginning students grasp of logic. The third problem in this book involves a complete analysis of the knights tour problem, for which the notation alone would stagger all but senior math majors. This text is intended for computer science majors.

Scheid, Francis

Introduction to Computer Science, Schaum's Outline Series, McGraw Hill Book Co., 1970, 281 pp., $4.95.

The fundamental question facing any beginner in computer science is just how is it that electricity can be trained to such a spectacular level of performance. In this book you will find the answer in two parts, hardware and software. Circuits, memory units, reading and writing devices are presented in chapters one through four. Programming is presented in chapters five through fourteen.

There seem to be several ways in which this book can be utilized: 1) a general introduction to computer science by using the whole book, 2) a short course in programming by using chapters seven and eight, 3) a longer course in programming by using chapters seven to fourteen.

This Schaum's Outline Series provides a host of solved problems and supplementary problems at the end of each chapter.

Sterling, Theodore; and Pollack, S. V.

Computing and Computer Science: A First Course, Macmillan Co., New York, 1970, 414 pp. (From Computing Reviews, Feb. 1971)

The authors set out "to provide students with an orderly presentation of the fundamentals of computer science." They take the view that computer science is basically a study of languages and communication for the formulating of automatic procedures. The text, divided into fifteen chapters, begins with Turing machines, and continues with chapters on character codes, computer arithmetic, and introductory computer organization. The notion of algorithm is clarified further via flowcharting.

Machine language and assembler language are treated in two chapters, using a simple hypothetical computer. Having motivated a need for more powerful languages the authors introduce PL/1 in four chapters. Then language processing in terms of symbol tables, hashing macros, etc., in a very sketchy way. Metalanguages are introduced using a modified BNF. The final two chapters cover hardware (analog, digital, and hybrid) and supervisory systems.

Thr reviewer is willing to accept the author's goals and view of computer science, but concludes that after a good start in the first chapter, the text simply does not fulfill reasonable expectations. If one is dealing with a science, one expects pertinent references--there seem to be no references. When dealing with the fundamentals, one expects clear definitions and consistent usage of terms; the authors tend to use computer jargon somewhat loosely. The text frequently refers to "modern computing machines." No machine is ever explicitly identified (save Turing's). Yet "modern computing machines display instructions in hexadecimal notation," and they all seem to be byte-oriented (8 bits each) with 32 words.

The programming languages, FORTRAN and COBOL, each rate one sentence; ALGOL is not mentioned. When memory protection is discussed, the only technique presented is typical of an IBM/360, yet this is not stated. From time to time notions of speed and cost of computers are mentioned, but no speeds or costs for any computer are ver cited.

Walker, Terry M.

Introduction to Computer Science: An Interdisciplinary Approach, Allyn and Bacon Inc., Boston, 1972, 530 pp.

That computer science can be studied by students from all disciplines is the approach of this book. For the most part one year of high school algebra would be the only mathematics prerequisite. The book does include sections that require more mathematics than that.

This introductory book was written embracing the idea that computer science should focus on teaching the concepts associated with problem solving. Attempts were made not to obscure the concepts of problem solving by introducing them solely through the use of a computer programming language. Programming language supplements are available for use with this book.

Some of the interdisciplinary topics included are: business problem solving, problems in social sciences, problem solving in linguisitics as well as problem solving in computer science. There is also a section on the impact of the computer on society.

In chapter two the desk calculator is used as an analogy of how a computer works. HYCOMP a hypothetical machine-language is developed to assist in gaining insight into computer design and the way computers solve problems. At the end of each chapter there are an abundance of problems to use in learning the material.

SECTION B: PERIODICALS

by Fred Daniels and Dave Dubose

Introduction

The intent of this section is to present as complete a list as possible of periodicals which are primarily devoted to computing. Appended to the list is a partial list of periodicals which, though not primarily devoted to data processing, frequently or occasionally contain articles pertaining to computing and computer science. Where it has been possible, the source through which the periodical can be obtained, the frequency of publication, and the subscription price have been listed.

In many cases, your school librarian may be able to provide other sources through which a given periodical can be obtained at a more reasonable price. There are several general subscription brokers that are sometimes able to offer subscription rates which are better than those offered by the publishers. In many cases quotes can be obtained for issues which are not in their general catalogue. A few of these brokers are listed below:

West Coast Organization Plan Inc.
Educational Division
P. O. Box 5538
phone: 209-268-6700

Ebsco Subscription Service
681 Market Street
San Francisco, California 94105

Popular Subscription Service
P. O. Box 1566
Terre Haute, Indiana
(Zip code and phone not available)

Where is has been possible to inspect copies of a particular magazine, an attempt has been made to describe its general contents and its suitability for use by the classroom teacher. Some of the magazines are highly technical and of little use to any but the most sophisticated high school teacher and of no use to the average high school student. However, in these magazines, news items, articles, and illustrations might provide an insight into the more recent hardware and software developments. Periodicals may be useful to teachers directly and some may be directly useful to the students.

The following are suggestions for classroom uses of periodicals. It is hoped that teachers will find these suggestions useful in their teaching situation and in evaluating the suitability of the periodicals in the list.

Student Use

A few periodicals are written at a level that the average student will find meaningful. Indeed some are aimed directly at the students. Some of the ways periodicals might be used by and with students are:

1. Display an assortment of magazines and computer related newspapers in a prominent place. Hopefully some students will discover items which will stimulate their interest and motivate them to initiate class discussions concerning these items.
2. Students can be assigned to read pertinent articles. Occasionally they can be assigned to give oral presentations to the class or write reports. (Try not to make this too much of a chore for them.)

3. Recent news items or articles on new developments, even pictures of a variety of types of hardware should be pointed out by the teacher. Such action might motivate students to read the magazines.
4. The teacher can read selected articles to the class (keep them short) and follow up with class discussions of the ideas presented. Items which would particularly lend themselves to this type of presentation are recent news items, articles concerning social implications, programs, computer applications and vocational opportunities.
5. Particularly appropriate articles can be duplicated and distributed to the students (with the publishers permission, of course).
6. Some periodicals or articles which are too technical for general student use could be used to provide enrichment materials for more capable students.

Teacher Use

Most of the available periodicals are written at a level too difficult for average students. Many are extremely technical and most well trained teachers would have difficulty understanding some of the articles. Each teacher will need to evaluate which ones are best suited to his situation. Some suggested uses for these periodicals are:

1. The informed teacher will want to keep abreast of news events and new developments in the computer field. Technical developments now may be important to the

teacher in the future. New developments might suggest new curriculum ideas.

2. New and/or different peripherals and communication equipment should be considered for future implementation. As new mini-computers are developed and prices come down they might be considered for implementation in the classroom.

3. Particular items which are too technical in their original form might be abstracted into a form more palatable to the students. They can then be either read and discussed in class or duplicated and distributed to the students.

4. Ideas for bulletin board or classroom displays can be found in periodicals. Free or inexpensive materials suitable for displays are advertised occasionally. Sometimes pictures, articles or cartoons can be cut out or reproduced for composing motivating or stimulating displays.

Computer Periodicals

Many of the periodicals in this list are available free (marked with **) from various agencies. For those publications which are "free to qualified subscribers", no attempt has been made to determine who is qualified. Hopefully educators will be considered qualified. The symbol T is used to indicate a publication which is recommended for use by teachers and the symbol S is used for those judged suitable for use by students. Only those publications which were personally inspected and for which a clear cut recommendation could be made were rated in this manner. Where the number of pages is indicated, this number is an average taken from random copies. Most of the publishers will furnish a sample copy on request.

ADP NEWSLETTER

Management Science Publishing, 430 Park Ave., New York, N.Y. 10022. $48 year, bi-weekly, 4 pages, general interest-management report.

AEDS JOURNAL

Association for Educational Data Systems, 1201 Sixteenth St., N.W., Washington, D.C. 20036. Free to members, $10 year to others, quarterly, 62 pages, special interest-education.

AEDS MONITOR

Association for Educational Data Systems, 1201 Sixteenth St., N.W., Washington, D.C. 20036. Free to members, $15 year to others, monthly, 24 pages, special interest-education.

AMERICAN JOURNAL OF COMPUTER APPLICATIONS

Society for Scholarly Computer Applications, P.O. Box 1960, Newport Beach, Cal. 92660. $18 year, quarterly, general interest.

AUDITOR'S COMPUTER UPDATE DIGEST

Professional Update Co., P.O. Box 36, Oakton, Virginia 22124. $26 year, loose-leaf book with monthly updating service, special interest-auditing.

AUTOMATED EDUCATION HANDBOOK

Automated Data Processing Newsletter, Management Science Publishing Co., 430 Park Avenue, New York, N.Y. 10022. $37.50 year, semimonthly, 4 pages, general interest.

AUTOMATION EXPRESS

International Physical Index, 1909 Park Ave., New York, N.Y. 10035. $110 year, monthly, 86 pages, American publication in English of current technical Russian literature dealing with automation topics.

AUTOMATED DATA PROCESSING IN STATE AND LOCAL GOVERNMENT **

Michigan Department of Education, Bureau of Library Services, 735 E. Michigan Ave., Lansing, Michigan 48913. Free, monthly, 8 pages, review of various journal and magazine articles.

BUSINESS AUTOMATION **

Hitchcock Publishing, Hitchcock Building, Wheaton, Illinois 60187. Free to qualified subscribers, $20 year to others, monthly, 64 pages, business-management oriented.

BUSINESS AUTOMATION NEWS ANALYSIS

Hitchcock Publishing, Hitchcock Building, Wheaton, Illinois 60187. Free to qualified subscribers, $25 to others, semi-monthly, 21 pages, general interest news analysis.

BUSINESS SOFTWARE INFORMATION SERVICE

The Business Press, 288 Park Avenue West, Elmhurst, Illinois 60126. $50 year, loose-leaf book with monthly updating service, special interest software.

COMMUNICATIONS OF THE ASSOCIATION FOR COMPUTING MACHINERY

Association for Computing Machinery, 211 E. 43rd St., New York, N.Y. 10017. Free to members. $35 year to others, monthly, 88 pages, primarily of technical interest with very little of general interest, contains important articles at edge of field, but doubtful use at high school level.

COMPUTER CHARACTERISTICS QUARTERLY

Adams Associates, 128 The Great Road, Bedford, Massachusetts 01730. $250 year, quarterly, 250 pages, provides data on the significant characteristics of over 360 central processors and peripheral devices.

COMPUTER DECISIONS **†

Hayden Publishing, 50 Essex St., Rochelle Park, New Jersey 07662. Free to qualified subscribers, $24 year to others, monthly, 48 pages, general-interest periodical, contains a nice news section and new hardware section, some articles at high school level.

COMPUTER DESIGN

Computer Design Publishing Co., 221 Baker Ave., Concord, Mass. 01742. Free to qualified subscribers, $15 year to others, monthly, 120 pages, special interest-electronics.

COMPUTER DEVICES

Hitchcock Publishing, Hitchcock Building, Wheaton, Illinois 60187. $95, 800 page loose-leaf book with monthly updating service, special interest-input/output devices.

COMPUTER DIGEST

$36.00 per year, monthly, special interest-computer graphics and image processing.

COMPUTER JOURNAL

British, $26 a year, quarterly.

COMPUTER NEWSFRONT

Computer Newsfront, 280 E. Main Street, Marlboro, Mass. 01752. $36 year, weekly, 16 pages, general interest.

COMPUTER NOTEBOOK

Auerback Corp., 121 N. Broad St., Philadelphia, Pennsylvania 19107. $175 year, loose-leaf book with bi-monthly updating service, guide to computer system characteristics.

COMPUTER POST CARD **

Computer Post Card Press, 866 United Nations Plaza, Suite 496, New York, N.Y. 10017. Free monthly, 16 pages, general interest. Advertising in the form of post cards.

COMPUTER PROGRAM ABSTRACTS

U.S. Government Printing Office, Superintendent of Documents, Washington, D.C. 20402. $1 per year, quarterly, general interest (catalog no. NAS 1.44).

COMPUTER PROCESSING UPDATER

Computer Research Bureau, 500 Newport Center Drive, Newport Beach, Calif. 92660. $55 year, semi-monthly, 8 pages, special interest-management.

COMPUTER SERVICES

Association of Data Processing Service Organizations, 825 S. Barrington, Los Angeles, California 90049. Free to members, $40 to others, bi-monthly, 32 pages, special interest-service centers.

COMPUTER STOCKS TODAY

Computer Research Bureau, 500 Newport Center Drive, Newport

Beach, Calif. 92660. $85 year, semi-monthly, 16 pages, special interest-financial information.

COMPUTER STUDIES IN THE HUMANITIES & VERBAL BEHAVIOR

COMPUTERS AND PEOPLE AND AUTOMATION

Berkeley Enterprises, 815 Washington St., Newtonville, Mass. 02160. $9.50 year, $18 two years, monthly, 50 pages, general interest, name will be changed Jan. 1, 1974 to "Computers and People" to reflect a newer emphasis, "The Computer Directory and Buyers Guide" is published yearly at $9 a copy, one of the few readable magazines at the high school level, articles are keyed in front, some articles appear to be unrelated to computer science.

COMPUTERS

The Institute of Electrical and Electronic Engineers, 345 East 47 St., New York, N.Y. 10017. Free to members IEEE, price available to non-members on request, monthly, 76 pages, technical journal.

COMPUTERS AND BIOMEDICAL RESEARCH

Academic Press, 111 Fifth Ave., New York, N.Y. 10003. $44 year, monthly, 88 pages, special interest-biomedical.

COMPUTERS AND HUMANITIES

Queens College of the City University of New York, Flushing, N.Y. 11367. $10 individuals, $20 institutions, 5 times a year, 64 pages, special interest-humanities, includes a little music, history and political science, archeaology, art, but mostly language and literature, most of the articles listed are reviews of articles in other periodicals, especially foreign periodicals.

COMPUTERWORLD

Computerworld, 129 Mt. Auburn St., Cambridge, Massachusetts 02138. $9 year, weekly, 42 pages, general interest, newspaper, plenty of news from government and industry, sections on communications, financial, software service, systems/peripherals, and small education section, interesting advertising (jobs wanted).

COMPUTING

COMPUTERS IN BIOLOGY & MEDICINE

Quarterly, $35 year.

THE COMPUTERIZED SOCIETY

COMPUTING NEWSLETTER FOR COMMUNITY COLLEGES

University of Colorado, Box 9630, Colorado Springs, Colorado 80909. $11 year, 9 issues per year, special interest-instructors in junior colleges.

DATA SYSTEMS

$11.10 year, monthly.

DATA SYSTEMS NEWS **

United Business Publications, P.O. Box 7387, Philadelphia, Pennsylvania 19101. Free to qualified subscribers, $10 to others, 40 pages, articles tend to be short and zappy, business and hardware emphasis, slick format, good news section.

DATA REPORTMETEOROLOGICAL ROCKET NETWORK FIRINGS

$36 year, monthly

DATA SYSTEMS NEWSTAPE

Media Horizons, 200 Madison Ave., New York, N.Y. 10016. $210 year, weekly news service on cassette tape, general management interest.

DATAMATION **T

Technical Publishing Co., 35 Mason Street, Greenwich, Conn. 06830. Free to qualified subscribers, $35 to others, bi-monthly, 160 pages, not too technical to read, includes technical, management and commentary sections, artful covers.

DATAMATION EXECUTIVE NEWSLETTER **

Technical Publishing Co. 35 Mason Street, Greenwich, Conn. 06830. Free to qualified subscribers, monthly 4 pages, general interest.

DIGITAL COMPUTER NEWSLETTER

Information Systems Branch, Office of Naval Research, Washington, D.C. 20360. Free to qualified subscribers, quarterly, 26 pages, general interest.

EDP ANALYZER

EDP Analyzer, 925 Anza Ave., Vista, Calif. 92083. $36 year, monthly, 16 pages, general interest bulletin.

EDP DAILY

EDP News Services, 514 Tenth Street, N.W., Washington, D.C. 20004. $250 year, daily (week-days), 8 pages, general interest.

EDP INDUSTRY REPORT & MARKET REVIEW

International Data Corp., 60 Aunstin St., Newtonville, Mass. 02160. $75 year, semi-monthly, 8 pages, general interest.

COMPUTING NEWSLETTER FOR INSTRUCTORS OF DATA PROCESSING

University of Colorado, Box 9650, Colorado Springs, Colorado 80909. $11 year, 9 issues per year, 10 pages, special interest education.

COMPUTING NEWSLETTER FOR SCHOOLS OF BUSINESS

University of Colorado, Cragmor Road, Colorado Springs, Colorado 80907. $18 year, 9 issues per year, 10 pages, special interest-faculties of schools of business.

COMPUTING REVIEWS

Association for Computing Machinery, 211 E. 43rd St., New York, N.Y. 10017. Free to members, $35 year to others, monthly, 102 pages, provides critical evaluations of books, papers, films and and video tapes on every aspect of computing, few of the articles are high school level, best way to see what is available in periodicals on a particular subject, however many of the periodicals reviewed are unknown and unobtainable, divided nicely into sections (one miniscule section relating to high school, junior high, and elementary).

COMPUTING SURVEYS

Computing Surveys, ACM Membership Dept. 1133 Ave. of Americas, New York, N.Y. 10036. ACM members $7, others $40, quarterly, 78 pages, general interest, articles assume limited exposure to computer science, this is one of the best periodicals for high school teachers, the articles are readable, assume little prior knowledge and yet go deep.

CRB BROKER SPECIAL

Computer Research Bureau, 500 Newport Center Drive, Newport Beach, Calif. 92660. $55 year, semi-monthly, 8 pages, special interest-financial information.

CYBERNETICA

DATA

$4 year, 3 issues year.

DATA AND CONTROL

$11.20 year, monthly.

DATA COMMUNICATIONS REPORT

Auerbach Corp., 121 N. Broad St., Philadelphia, Pennsylvania 19107. $325 initial year, $250 year thereafter, loose-leaf book with quarterly updating service, guide to computer support equipment.

THE DATA COMMUNICATIONS USER

Communications Trends, 181 South Franklin Ave., Valley Stream, N.Y. 11581. Free to qualified individuals, $20 year to others, monthly, 64 pages, special interest-telecommunications.

DATA AUTOMATION

$.5 year, monthly, attractive format, articles generally readable, interesting hardware information.

DATA DAILY

$150 year.

DATA MAGAZINE

$12 year, monthly.

DATA MANAGEMENT

Data Processing Management Association, 505 Busse Highway, Park Ridge, Illinois 60068. Free to members, $5 year to others, monthly, 68 pages.

DATA PROCESSING

North American Publishing Co., 134 N. 13th St., Philadelphia, Penn., 19107. $8.50 year, monthly, 76 pages, general interest.

DATA PROCESSING DIGEST

Data Processing Digest, 1140 S. Robertson Blvd., Los Angeles, Calif., 90035. $36 year, monthly, 20 pages, general interest.

DATA PROCESSING MAGAZINE

$10 year, monthly.

DATA PROCESSING FOR EDUCATION

American Data Processing, 4th Floor, Book Building, Detroit, Michigan 48226. $24 year, monthly, 12 pages, special interest-education, discusses mutual interests in education, disseminates current information pertinent to increasing data processing in the education field.

DATA PROCESSING FOR MANAGEMENT

American Data Processing, 22nd Floor Book Tower, Detroit, Michigan 48226. $8.50 year, monthly, 56 pages, general interest.

DATA PROCESSOR

EDP WEEKLY

EDP News Services, 514 Tenth Street, N.W. Washington, D.C. 20004. $75 year, weekly, 16 pages, general interest.

EDUTRONICS NEWSLETTER **

Edutronics Systems International, 3345 Wilshire Blvd., Los Angeles, Cal. 90005. Free to qualified subscribers, monthly, 4 pages, special interest-education.

FAST ANNOUNCEMENT SERVICE

National Technical Information Service, U.S. Department of Commerce, Springfield, Virginia 22151. $5 year, 2 pages, frequency depends on input highlights selected from New Government Research development reports.

FASTLINE MONTHLY

International Data Publishing Co., P.O. Box 1, Newtonville, Mass. 02160. $38.50 year, monthly, 8 pages, marketing intelligence on new installations.

GOVERNMENT DATA SYSTEMS TS

United Business Publications, $6 year, bi-monthly, 46 pages, contains a good selection of general interest articles, covering current computer usage.

GOVERNMENT REPORTS TOPICAL ANNOUNCEMENTS

National Technical Information Service, U.S. Department of Commerce, Springfield, Virginia 22151. $5 year, semi-monthly, 8 pages, abstracts of recently released reports resulting from government-sponsered research and development efforts.

HONEYWELL COMPUTER JOURNAL **

Honeywell, Inc., Electronic Data Processing, Division, Wellesley Hills, Mass. 02181. Free, quarterly, 54 pages, technical interest.

HONEYWELL EDP NEWSLETTER **

Honeywell, Inc., Industry Marketing, 60 Walnut Street, Wellesley, Mass. 02181. Free, monthly, 4 pages, general interest.

HOSPITAL/MEDICAL AUTOMATION HANDBOOK

American Data Processing, Inc., 19830 Mack Ave., Detroit, Michigan 48236. $199 year, 1,000 page loose-leaf book with monthly updating service, special interest-medical.

IBM COMPUTING REPORT TS**

International Business Machines Corp., Data Processing Division, 1133 Westchester Ave., White Plains, New York 10604. Free, 5 issues per year, 17 pages, attractive format, approx. half of articles non-technical, IBM hardware and software usage emphasized, contains several outstanding articles per issue that could be used in high school.

IBM DATA PROCESSOR **

International Business Machines Corp., Data Processing Division, 112 E. Post Road, White Plains, New York 10601. Free, monthly, 34 pages, general interest.

IBM JOURNAL OF RESEARCH & DEVELOPMENT

International Business Machines Corp., Armonk, New York 10504. $7.50 year, bi-monthly, 84 pages, highly technical research journal.

IBM SYSTEMS JOURNAL

International Business Machines Corp., Armonk, New York 10504. $5 year, quarterly, 88 pages, special interest-systems.

INTERFACE **

Auervack Corp., 121 N. Broad Street, Philadelphia, Pennsylvania. Free to qualified subscribers, quarterly, 4 pages, general interest.

INTERNATIONAL JOURNAL OF COMPUTER MATHEMATICS

Gordon & Breach, 440 Park Ave. S., New York 10016. $12 year, quarterly, 108 pages, technical interest.

INTRODUCTION TO COMPUTING AND COMPUTER SCIENCE

JOURNAL OF COMPUTER & SYSTEM SCIENCES

Academic Press, Inc., 111 Fifth Ave., New York, N.Y. 10003. $12 year, quarterly, 108 pages, technical interest.

JOURNAL OF DATA EDUCATION TS

Journal of Data Education, 76 Union, Northfield, Vermont 05663. Free to SDE members ($12 dues), $10 yearly to institutions, 8 issues a year, 30 pages, special interest, computers in education, programming emphasis, articles at both student and teacher level.

JOURNAL OF EDUCATIONAL DATA PROCESSING

Educational Systems Corp., Box 3711, Georgetown Station, Washington, D.C. 20007. $9 year, quarterly, 142 pages, special interest education.

JOURNAL OF SYSTEMS MANAGEMENT

JOURNAL OF THE ASSOCIATION FOR COMPUTING MACHINERY

Association for Computing Machinery, 211 E. 43rd Street, New York, N.Y. 10017. Free to members, $30 to others, quarterly, 168 pages, technical interest, much too technical for high school level.

MANAGEMENT CONTROLS

MICROFILM AND COPIERS

Hitchcock Publishing Co., Hitchcock Building, Wheaton, Illinois 60187. $75 year, 300 page loose-leaf book with monthly updating service, general interest.

MODERN DATA **T

Modern Data Services, 3 Lockland Ave., Framingham, Mass. 01701. Free to qualified subscribers, $18 year to others, monthly, special interest-management, interesting news articles, new products, "what hath babbage wrought", good for student discussion.

MOODY'S COMPUTER INDUSTRY SURVEY

Brandon Applied Systems, Inc., 1700 Broadway, New York, N.Y. 10019. $110 year, monthly, 32 pages, special interest-management.

NEWS BULLETIN **

Business Equipment Manufacturers Association, 235 E. 42nd Street, New York, N.Y. 10017. Free to qualified subscribers, weekly, 6 pages, general interest.

PEOPLE'S COMPUTER COMPANY TS

People's Computer Company, P.O. Box 310, Menlo Park, Cal. 94205. $2 year for students, $4 for others, 5 issues a year, 24 pages newspaper form, plenty of mod artwork, plenty of Basic language programs, definitely aimed at the high school student.

PERIPHERALS WEEKLY

EDP News Services, 514 Tenth Street N.W., Washington, D.C. 20004. $75 year, weekly, 16 pages, special interest.

PRACTICAL METHODS FOR SCHOOLS **

Automated Business Systems, Division of Litton Industries, 600 Washington Ave., Carlstadt, New Jersey 07072. Free, quarterly, 4 pages, education.

QUARTERLY BIBLIOGRAPHY OF COMPUTERS AND DATA PROCESSING

Applied Computer Research, 8900 N. Central Ave., Suite 208, Phoenix, Arizona 85020. $29.50 year, quarterly, general interest.

REAL TIME

Control Data Corp., 8100 34th Ave., S., Minneapolis, Minnesota 55440. Free, bi-monthly, 8 pages, general interest.

REPROGRAPHICS & ENGINEERING RECORDS MANAGEMENT

Reprographics, 750 Third Ave., New York, N.Y. 10017. $18 year, monthly, 36 pages, special interest, graphics journal.

SCIENTIFIC & CONTROL COMPUTER REPORTS

Auerback Corp., 121 N. Broad St., Philadelphia, Pennsylvania 19107. $500 initial year, $240 year thereafter, loose-leaf book with bi-monthly updating service, guide to scientific and control computers.

SIMULATION

Simulation Councils, P.O. Box 2228, La Jolla, Calif. 92307. $28 year, monthly, 58 pages, special interest-computer simulation.

SIMULATION GAMING NEWS

Simulation Gaming News, Box 8899, Stanford University, Stanford, California 94305. $4 year, 5 times year, newspaper format, 16 pages, usually contains one featured game at elementary or high school level, also technical information.

SOFTWARE

Hitchcock Publishing Co., Hitchcock Building, Wheaton, Illinois 60187. $85 year, 600 page loose-leaf book with monthly updating service, special interest-software.

SOFTWARE AGE **

Press Tech, 1020 Church St., Evanston, Illinois 60201. Free to qualified subscribers, $10 year to others, bi-monthly, 18 pages, general interest.

SOFTWARE DIGEST

EDP News Services, 514 Tenth Street, N. W., Washington, D.C. 20004. $75 year, weekly, 16 pages, general interest.

SOURCE DATA AUTOMATION REPORT

Information Spectrum, 1020 Kings Highway North, Cherry Hill, New Jersey 08034. $95 year, loose-leaf book, quarterly updating service, cost/performance descriptions of commercially available equipment.

STANDARD EDP REPORTS

Auerbach Corp., 121 N. Broad St., Philadelphia, Pennsylvania 19107. $900 initial year, $750 year thereafter, loose-leaf books with 10 updating services per year, appraisal of today's EDP systems.

SYSTEMATION LETTER

Systemation, Box 730, Colorado Springs, Colorado 80901. $32 year (includes one index annually, three hard bound reference handbooks, and one bonus handbook annually), monthly, 6 pages, general interest.

TECH FILES

Modern Data, 3 Lockland Ave., Framingham, Mass. 10701. There are 12 Tech File subjects: Interactive CRT Displays, Printers & Teleprinters, Minicomputer, Disk/Drum Memories, Cassette-Cartridge Transports, OCR & Mark Readers, Data Sets & Multiplexers, COM Equipment, Key-to Tape/Disk, Magnetic Tape Transports, Digital Plotters, and Time Sharing Services, Each Tech File is $60 per year, loose-leaf with quarterly updating service.

TWO BITS WORTH **

H. B. Maynard & Company, Management Science Division, 718 Wallace Ave., Pittsburgh, Pa. 15221. Free, monthly, 4 pages, general interest.

UPDATE FOR MANAGEMENT ENGINEERS **

Sperry Rand Corporation, UNIVAC Division, 1290 Avenue of the Americas, New York, N.Y. 10019. Free, bi-monthly, 4 pages, general interest.

OTHER PERIODICALS WHICH OCCASIONALLY CONTAIN ARTICLES RELATED TO COMPUTING

AMERICAN EDUCATION

AMERICAN JOURNAL OF PHYSICS

ARITHMETIC TEACHER

AUDIOVISUAL INSTRUCTION

BUSINESS WEEK

CLEARING HOUSE

CONTEMPORARY PHYSICS

CURRENT SCIENCE

DISCRETE MATHEMATICS

EARTH SCIENCE

EDUCATION CANADA

ELECTROCHEMICAL SOCIETY JOURNAL

ELECTRONIC ENGINEERING

ELECTRONIC NEWS

ELECTRONICS

ENGINEERING GEOLOGY

ENGLISH JOURNAL

MATHEMATICS TEACHER

NATIONS SCHOOLS

SCHOOL LIBRARIES

SCHOOL MANAGEMENT

SCIENCE

SCIENTIFIC AMERICAN

SECTION C: NONPRINT MATERIALS

Nonprint matetials are materials which are primarily visual in nature. They are usually projected rather than printed. Some consider printed materials that are formed into transparencies and projected to be nonprint material. Any additional material which shows the working parts, illustrates, or can be manipulated, other than books, would be considered nonprint material. Some examples of nonprint materials would include: films, film strips, audio tapes, video tapes, film loops, transparencies, slides, computer delivered materials, charts, graphs, and manipulative aids.

An excellent catalog of nonprint materials can be obtained from Mr. Ben Jones at OTIS. The following is a description of that catalog.

The CATALOG OF NONPRINT MATERIALS USEFUL IN COMPUTER RELATED INSTRUCTION is a bibliography of non-books materials for teachers in all levels of public instruction. Every teacher, whether public school or college, will find some audio-visual materials listed in this catalog to be of use in their instruction.

Nearly 200 items are alphabetically arranged (by title) displaying type of material, and other annotative information. A four letter code indicates where the material may be obtained. The code is keyed to a source list with addresses. The alphabetical listing of materials is indexed by subject headings and is cross referenced.

Items in the catalog have been culled from computer vendor materials, professional association bibliographies, and Oregon instructional materials centers. (No price quotation is available at this time.)

Copies of the catalog may be obtained by writing:

Ben Jones
Supervisor, Instructional Services
Oregon Total Information System
345 East 40th
Eugene, Oregon 97405

SECTION D: PLACES TO VISIT

This section lists both places for the teacher and the students to visit. Places for teachers to visit are involved in computer education. Places for students to visit are suitable for field trips.

It is recognized that this list is not exhaustive. Some important places may have been overlooked, if so we both want to offer our apology and we would like to be notified of this omission so that it can be corrected in the future.

TEACHERS

When a teacher is considering the development of a computer science, computer programming, or computer literacy course it can be helpful to see a similar course in actual operation. This part is designed to give you a clue to where activity is occuring, and where you might go to see things of special interest.

COMPUTERS IN THE MATHEMATICS PROGRAM

Benson High School
Mr. Carl Bryson
Portland, Oregon

Rex Putnam High School
Mr. Wally Rogelstad
Milwaukie, Oregon

Willamette High School
Mr. Bill Best
Eugene, Oregon

Wilson High School
Mr. Joe Brigatto
Portland, Oregon

COMPUTER SCIENCE/COMPUTER LITERACY

Ashland High School
Mr. Keith Garrett
Ashland, Oregon

Centennial High School
Mr. Fred Daniels
Gresham, Oregon

Reynolds High School
Mr. Earl Phillips
Troutdale, Oregon

Sam Barlow High School
Mr. Dave Dubose
Gresham, Oregon

SMALL (in school) COMPUTER SYSTEMS

Newport High School
Mr. Dave Dempster
Newport, Oregon

McMinville High School
Mr. Bill Peterson
McMinville, Oregon

COMMUNITY COLLEGE COMPUTING

Blue Mountain Community College
Mr. Wally Waldman
Pendleton, Oregon

Lane Community College
Mr. John Loughlin
Eugene, Oregon

Portland Community College
Mr. Tom Crowder
Portland, Oregon

Southwestern Oregon Comm. College

Coos Bay, Oregon

OTHER PLACES OF INTEREST

Mount Hood Community College
Dr. Diane Dean
Gresham, Oregon

Multnomah County IED
Mr. Jack Allen, Jack Slingerland
Portland, Oregon

Oregon Museum of Science and Industry
Mr. Rusty Whitney
Portland, Oregon

Oregon State University
Dr. Ed Anderson, Jo Ann Baughman
Corvallis, Oregon

Oregon Total Information System
Mr. Ben Jones, Mike Neill
Eugene, Oregon

Rogue Valley Council on Computer Education
Mr. Keith Garrett
Ashland High School
Ashland, Oregon

University of Oregon
Dr. David Moursund
Eugene, Oregon

STUDENTS

The following is a list of some interesting places for a student to visit.

Pacific Northwest Bell Telephone
Contact local office

International Business Machines
Portland, Oregon

Tektronix
Beaverton, Oregon

Household Finance--Loan Co.
Eugene, Oregon

Oregon Total Information System
Eugene, Oregon

Bonneville Power Administration
Portland, Oregon

State Department of Labor
Salem Oregon

Oregon Museum of Science and Industry
Portland, Oregon

Multnomah County IED
Portland, Oregon

ERIC
University of Oregon--Eugene

Southern Oregon College
Ashland, Oregon

United Air Lines
Portland

SECTION E: KEY PEOPLE

Educational use of computers is fairly well developed in Oregon High Schools. A number of people have distinguished themselves in this development. The following lists some of the key people who are currently available in Oregon. Several others who have had a significant effect upon computers in education such as Judy Edwards, Don Roznagel and Jim Norton, have gone to work in other states.

The people on this list were selected because they have been involved in computer education for many years, are recognized leaders in this field and are excellent sources of additional information.

MIKE DUNLAP
Department of Computer Science
University of Oregon
Eugene, Oregon

KEITH GARRETT
Ashland High School
Ashland, Oregon

TIM KELLEY
Computer Center
Southern Oregon College
Ashland, Oregon

DAVID MOURSUND
Department of Computer Science
University of Oregon
Eugene, Oregon

MIKE NEILL
Lane County IED
Eugene, Oregon

JACK SLINGERLAND
Multnomah County IED
Portland, Oregon

RUSTY WHITNEY
Oregon Museum of Science and Industry
Portland, Oregon

A list of Oregon educators involved in computer education can be obtained by joining the OREGON COUNCIL FOR COMPUTER EDUCATION (OCCE). Write to: Oregon Council for Computer Education; 4015 S.W. Canyon Road, Portland, Oregon 97221

SECTION F: MAJOR COMPUTER-EDUCATION PROJECTS

This section contains the names and addresses of major computer-education projects. It includes sections on: college sponsored consortiums, area school district sponsored consortiums, school sponsored consortiums, commercial consortiums, and on-going instructional computer projects.

This information was obtained in part from "The Use of Computers in Instruction in Secondary Schools", Warren J. Koch, National Association of Secondary School Principals, 1201 Sixteenth Street, N. W., Washington, D.C. 20036.

COLLEGE SPONSORED CONSORTIUMS

SOLO
Thomas Dwyer, Director
Pittsburgh, Pa. 15213

University of Rhode Island
William Hemmerle, Director
Kingston, R.I. 02881

AREA SCHOOL DISTRICT SPONSORED CONSORTIUMS

METCOM
Jack Allen, Director
Box 16657
Portland, Oregon

OTIS (Oregon Total Information System)
Bob Dusenberry, Director
345 E. 40th
Eugene, Oregon

TIES (Total Information for Educational Systems)
Thomas Campbell, Director
1925 W. County Road B-2
St. Paul, Minn. 55113

Region IV Education Service Center
T. S. Handcock, Director
Houston, Texas 77002

LOCAL
Robert Haven, Director
44 School Street
Westwood, Mass. 02090

LIRICS (the Long Island Regional Instructional Computer System)
Gerard Dann, Director
17 Westminster Avenue
Dix Hills, N. Y. 11746

EKEDC (East Kentucky Educational Development Corporation)
Edwin Jones, Director
325 Winchester Avenue
Ashland, Ky. 41101

SCHOOL SPONSORED CONSORTIUMS

South Portland, (Me.) High School
Ann Waterhouse
637 Highland Avenue
South Portland, Me. 04106

Wayne (N.J.) Consortium
Henry Peterson
Wayne Hills High School
Wayne, N. J. 07470

COMMERCIAL CONSORTIUMS

PLAN (Program for Learning in Accordance with Needs)
Westinghouse Learning Corporation
235 Wyman
Waltham, Mass 02154

Time Share Corporation
Hanover, N. H. 03755

ON-GOING INSTRUCTIONAL COMPUTER PROJECTS

INDICOM
Donald Arnold, Director
Waterford Township School District
1325 Crescent Lake Road
Pontiac, Mich. 48054

REFLECT
William M. Richardson, Director
Albert Einstein High School
11135 Newport Mill Road
Kensington, Md. 20795

CATER
Richard Haskell, Director
Dawnwood Junior High School
Centerreach, Long Island, N. Y. 11720

Philadelphia (Pa.) School System
Sylvia Charp, Director
Philadelphia School District
Philadelphia, Pa. 19104

Buffalo, N. Y.
Robert Santuci
Computer Science Department
Hutchinson Central Technical High School
256 South Elmwood Ave.
Buffalo, N. Y. 14222

Half Hollow Hills
Gerard Burke, Director
Half Hollow Hills High School
50 Vanderbilt Parkway
Dix Hills, N. Y. 11746

REACT (Relevant Educational Applications of Computer Technology)
Dick Lynch
S. W. Second Avenue
Portland, Oregon

There is no complete source of on-going projects. Additional information can be obtained from National Association of Secondary School Principals, Committee on Educational Technology, 1201 Sixteenth Street, N. W., Washington, D. C. 20036