ED 273 268 IR 012 282

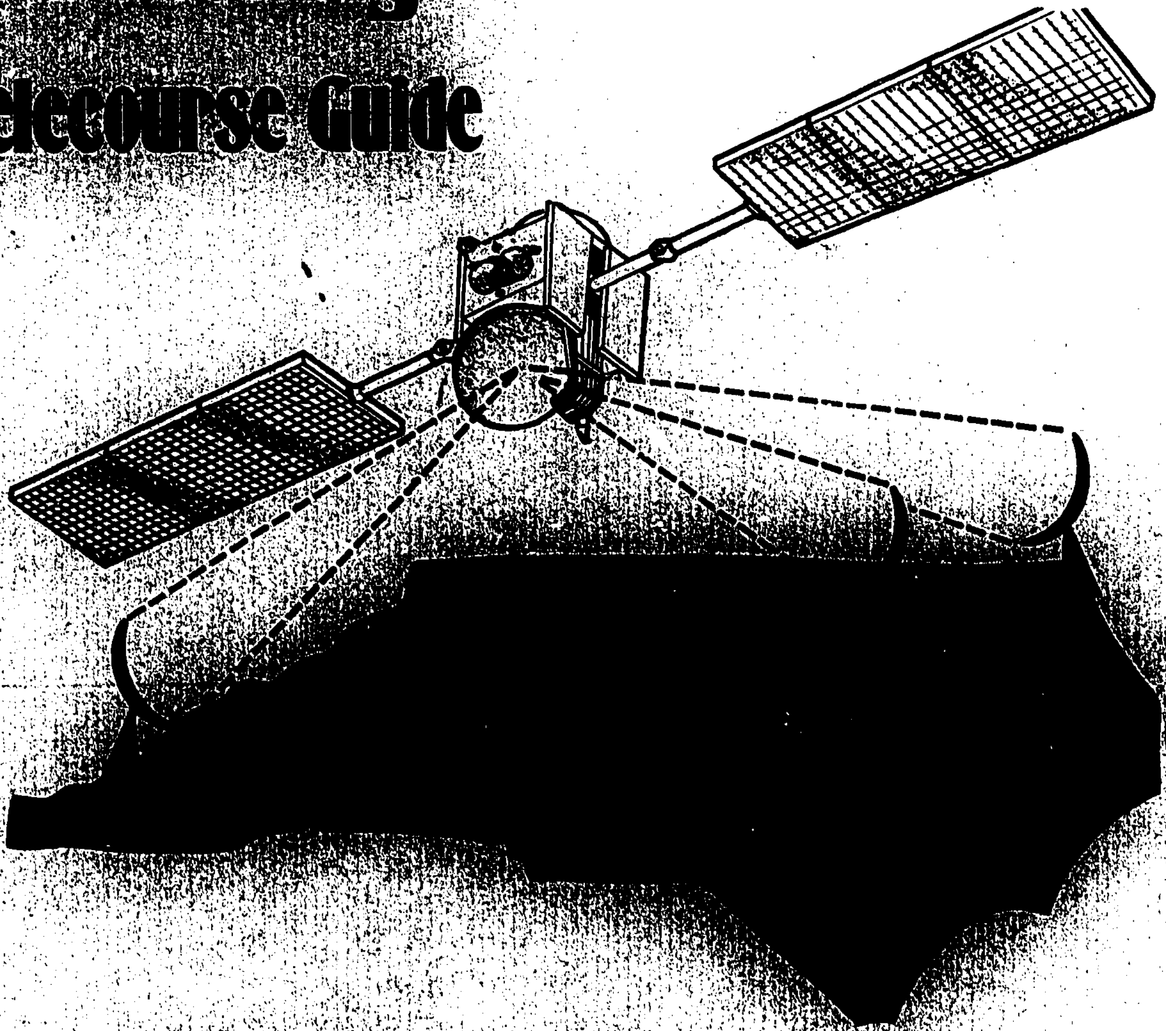
TITLE	Using Computers in Teaching: Telecourse Guide. A Pilot Project in Distance Learning by Satellite.
INSTITUTION	North Carolina State Dept. of Public Instruction, Raleigh. Div. of Educational Media.
PUB DATE	86
NOTE	100p.; This pilot project was tested in weekly 1-hour sessions between April 15 and May 20, 1986.
PUB TYPE	Guides - Classroom Use - Guides (For Teachers) (052)
EDRS PRICE	MF01/PC04 Plus Postage.
DESCRIPTORS	*Communications Satellites; *Computer Assisted Instruction; *Computer Software; *Distance Education; Elementary Secondary Education; Inservice Teacher Education; *Microcomputers; Pilot Projects; Study Guides; *Telecourses
IDENTIFIERS	*North Carolina

ABSTRACT

Part of a pilot project in distance learning by satellite in North Carolina, this telecourse is designed to increase teachers' awareness and understanding of how to use computers and instructional software in their teaching. The six 1-hour teleconference sessions in this telecourse include "live," interactive one-way video/two-way audio communications between the presenters in the studio and teachers in the three participating schools. This printed guide, the computers and two sets of software (some run on the IBM PC, others on the Apple IIe), and the telecourse facilitator complete the project's communications/instructional system. Structured as a survey of some basic and instructional applications of the computer, the telecourse includes an orientation to computer applications in the K-12 instructional program and sessions on writing; data analysis and problem solving; data collection, analysis, and graphing; arts education; and authoring and graphing. Subject areas represented in these sessions include communication skills, science, social studies, music, foreign languages, and mathematical sciences. Designed to engage participants in active involvement as they learn, this guide includes participant objectives, a presentation synopsis, teleconference preparation and follow-up activities, independent practice/personal growth activities, and a list of references and resources for each session. Detailed instructions for using individual software packages are also provided as appropriate, and some related uses of the software may be suggested. Most of the assigned readings are appended to the manual, including the final draft of the North Carolina competency-based curriculum guide for library/media and computer skills for grades K-12. (DJR)

Using Computers in Teaching

Telecourse Guide



A Pilot Project in Distance Learning by Satellite

April 15-May 20, 1986

Educational Media and Technology
Department of Public Instruction
Raleigh, North Carolina

USING COMPUTERS IN TEACHING

Telecourse Guide

Contents

INTRODUCTION

Distance Learning by Satellite

USING COMPUTERS IN TEACHING is a staff development telecourse being offered as part of the pilot project in distance learning by satellite. The project is conducted by the North Carolina Department of Public Instruction in cooperation with Appalachian State University and the Agency for Public Telecommunications.

The telecourse is being delivered by satellite from a studio on the campus of Appalachian State University through the University's Ku-band satellite uplink, AppalLink. The audio and video signals will be uplinked to the SBS III communications satellite and downlinked to the receiving earth stations (dishes) that have been installed for this project at Currituck High School, Roanoke Rapids High School, and Stecoah School.

The innovative features of the project include "live," interactive one-way video/two-way audio communications between the presenters in the studio and the participants in the three schools. Open telephone lines will make it possible for the participants to interact orally with the people they see and hear on television. Communication between receiving sites is also possible.

This printed guide, the computers and software packages, and the telecourse facilitator complete the project's communications/instructional system.

The goals of the pilot project are (1) to identify what satellite communications systems are appropriate for public school education, both for staff development and classroom instruction; (2) to determine the competencies needed to use a satellite communications system; (3) to identify the components of a successful satellite communications staff development model; (4) to evaluate the use of satellite communications for delivering instructional models and materials.

Telecourse Development

The major purpose of the telecourse is to increase teachers' awareness and understanding of how to use computers and instructional software in their teaching. It addresses educators' needs for information, models, and practice in effective uses of computers in instruction. The utilization of the computer software demonstrated in each teleconference relates to the North Carolina Standard Course of Study and the Teacher Handbooks for the competency-based curriculum. The telecours‿ is structured as a survey of some basic and instructional applications of the computer.

Telecourse Organization

USING COMPUTERS IN TEACHING consists of six one-hour teleconferences to be cond\cted weekly beginning April 15 and ending May 20, 1986.

The other course materials and activities are the guides, readings, practice in using the software that is demonstrated and discussed in the teleconferences, and the telecourse evaluations. The telecourse has been approved for one unit of renewal credit by the North Carolina Department of Public Instruction. This approval is based on nine contact hours, one and a half hours each week on the day of the teleconference, and two and a half hours of lab work, or practice with the software, between sessions. The telecourse facilitator will lead the group through the course and the teleconferences.

Telecourse Guide

The guide for each teleconference has been prepared by the instructors/ presenters to engage the participants in active involvement as they learn more about how to use computers in their teaching. To get the most from the teleconferences, the participants are expected to read the guide and do the teleconference preparation and follow-up activities. For the participants' convenience, most of the assigned readings are in the Appendices.

Teleconference Schedule/Presenters/Computer Software

The teleconferences are scheduled for 3:30-4:30 p.m. on the dates shown on the next page. The presenters are consultants from the Educational Media and Technology Area and the Instructional Services Area of the State Department of Public Instruction and the Department of Mathematics Science at Appalachian State University.

Two sets of the software used or mentioned in the teleconferences are available in each of the three participating schools. The titles of the software are indicated on the teleconference schedule. RESEARCH ASSISTANT and SNAPGRAPH run on IBM PC, the other software runs on Apple IIe.

USING COMPUTERS IN TEACHING

Teleconference Schedule
3:30 - 4:30 p.m.

Date	Topic	Presenters	Software
Tues., April 15	Telecourse Orientation	Margaret Bingham Computer Services Reta Richardson Television Services	
Wed., April 23	Writing	Cecilia Denning Computer Services Mike Frye Communication Skills	FrEdWriter
Tues., April 29	Data Analysis, Problem Solving	Doug Robertson Social Studies	Bank Street Filer Data Bases
Tues., May 6	Data Collection, Analysis, Graphing	Dr. William E. Spooner Science	Science Toolkit Snapgraph Research Assistant
Tues., May 13	Arts Education	Preston Hancock Music Doc McCulloch Creative Instructional Systems	Melodic Dictator Sebastian II Koalapainter with Koala Pad
Tues., May 20	Authoring, Graphing	Fran Hoch Foreign Languages Dr. Mark Harris Mathematical Sciences ASU	Brainz-Gamz Graphit

USING COMPUTERS IN TEACHING 1

Telecourse Orientation

Subject Area: Media and Technology/Computer Education
Grade Levels: K-12

PARTICIPANT OBJECTIVES

As a result of participating in the teleconference and completing the related activities, the participant will be able to:

1. Identify at least two ways the communications system used in this distance-learning-by-satellite project can be applied to the statewide implementation of the Basic Education Program, the Standard Course of Study, and the Teacher Handbook for the competency-based curriculum.

2. Explain to colleagues and students how this project is combining the technologies of video, audio, telephone, computers, print, satellite, and people to deliver "live" staff development training directly to three groups of teachers in Currituck County, Graham County, and Roanoke Rapids.

3. Identify several possible computer applications for solving relevant problems in the K-12 instructional program.

4. Discuss several teaching strategies necessary to integrate computer courseware into the on-going instructional program.

5. Explain the correlation between the computer skills objectives and specific instructional area skills objectives contained in the Teacher Handbook.

PRESENTATION SYNOPSIS

The teleconference will introduce the participants to this pilot project in distance learning by satellite and the telecourse it offers. It will include:

- Practice in using the telephone for the interactive two-way audio and one-way video participation between the presenters and the participants.

- Discussion of the purpose and rationale for this project and its possible applications to the implementation of the Basic Education

Progıam, the Standard Course of Study, and the Teacher Handbook for the competency-based curriculum.

- Examination of the components of the communications system: video, audio, telephone, satellite, print, computer, and people.
- Preparation of the participants for the telecourse and the subsequent five teleconferences on using the compute: in teaching, including matching types of software to curriculum areas and identifying related competencies in the Teacher Handbook.
- Discussion of teaching strategies necessary to integrate this software into the on-going instructional program.

TELECONFERENCE PREPARATION

1. Read the Basic Education Program for North Carolina's Public Schools and the Background and Overview in the North Carolina Standard Course of Study, pages 3-23.

TELECONFERENCE FOLLOW-UP

Review Questions:

1. What are the advantages of delivering instruction, training, and information through the communications system used in this pilot project? The disadvantages?
2. What types of computer courseware other than drill and practice programs exist that are appropriate for every subject and every grade level?
3. What is a teaching strategy necessary for integrating computer courseware into the instructional program that would not be necessary when using a filmstrip or videocassette?
4. What is the correlation between the computer skills objectives in the Teacher Handbook and the instructional area skills objectives?

INDEPENDENT PRACTICE/PERSONAL GROWTH ACTIVITIES

1. Read computer competency goals, objectives, and measures for grades K-12 in the Teacher Handbook - Library/Media & Computer Skills K-12. These are found in Appendix 1, Final Draft, Computer Awareness Competency Goals, Objectives, and Measures from the Teacher Handbook for Library/Media & Computer Skills K-12, North Carolina Competency-Based Curriculum.

2. Read Level I - Section 1.0 & 4.0 of Computer Competencies for All Educators.

3. View videocassette of "The Electronic Classroom," Virginia State Department of Education.

4. View videocassette of "German by Satellite," Oklahoma State University.

REFERENCES/RESOURCES

Computer Competencies for All Educators in North Carolina Public Schools, Department of Public Instruction, Raleigh, July, 1985.

North Carolina Basic Education Program for North Carolina's Public Schools, Raleigh, Revised November, 1985.

North Carolina Standard Course of Study and Introduction to the Competency-Based Curriculum, Department of Public Instruction, Raleigh, 1985.

Teacher Handbook - Library/Media & Computer Skills K-12, North Carolina Competency-Based Curriculum, Department of Public Instruction, Raleigh, 1985.

USING COMPUTERS IN TEACHING 2

Writing

Subject Area: Communications Skills
Grade Levels: 9-12

PARTICIPANT OBJECTIVES

As a result of participating in the teleconference and completing the related activities, the participant will:

1. Be aware of the basic features of a word processor.

2. Understand how to compose a writing tutorial through the use of guiding questions.

3. Be aware of the use of a word processor for generating teacher-made materials.

4. Be aware of how a computer can be used to prepare students for The Minimum Competency Writing Test.

PRESENTATION SYNOPSIS

The teleconference introduces the participants to the computer as a word processor that can be applied to any instructional activity that requires writing. The session will involve the participants in "hands on" practice with word processing as the instructors demonstrate the following:

- Loading, entering, inserting, deleting and printing text by using the prepared file, "Essay."

- Printing the first draft and making suggestions on how this document could be improved.

- Revising and editing, final printing, and saving the document.

- Modifying the file.

TELECONFERENCE PREPARATION

1. Review the criteria used to evaluate the minimum competency writing test in Writing Essay Test: A Teacher's Handbook.

2. Read the articles in Appendix 2.

3. Use FrEdWriter Tutorial to become familiar with basic word processing terms. Directions are printed at end of this guide.

4. Practice using FrEdWriter (refer to the printed directions).

5. Practice loading "Essay" file.

6. Practice entering, deleting, inserting, and printing text; practice moving cursor.

Study Questions

1. What capabilities does word processing have that facilitates the teaching of writing?

2. What capabilities of word processing are useful for generating teacher-made materials?

TELECONFERENCE FOLLOW-UP

Review Question

1. How would you use this guiding question technique to practice for other types of writing?

INDEPENDENT PRACTICE/PERSONAL GROWTH ACTIVITIES

1. Compose an original tutorial for The Minimum Competency Writing Test.

2. Compose three different versions of a ten-item quiz.

REFERENCES/RESOURCES

TEACHING WRITING: A PROCESS APPROACH, telecourse produced by Maryland ITV, distributed in North Carolina by School Television.

"Word Processing," READY OR NOT, School Television series, Department of Public Instruction, Raleigh.

Teacher Handbook - Communications Skills K-12, North Carolina Competency-Based Curriculum, Department of Public Instruction, Raleigh, 1985.

Writing Essay Test: A Teacher's Handbook, North Carolina Competency Test Program, Department of Public Instruction, Raleigh, January, 1986.

RELATED USES

Any subject area requiring written material.

DIRECTIONS FOR USING FrEdWriter

Put the disk in and turn computer and monitor on. The main program menu will come on the screen. Choose 2, "Start FrEdWriter." Then choose 80-column format by pressing 8.

Hit RETURN three times to get through credit screens. Under the light green line (top left) is a flashing cursor to mark the spot where you will begin your work.

[Stop at this point and use FrEdWriter Tutorial. Hold down CONTROL and Press T for the Tutorial. You may choose to practice using FrEdWriter at this time. If so, continue to follow the directions. If you wish to practice at another time, you must start at the beginning and skip the directions within the brackets.]

A file has been prepared for your practice. Retrieve the "ESSAY" file from the disk. To retrieve this file, hold down CONTROL and PRESS L for load. At the bottom of the screen where the cursor is flashing, type ESSAY (The name of the file) and PRESS RETURN. "ESSAY" file will load. To get to beginning of file, hold down CONTROL and Press B.

To move from question to question, Press the DOWN arrow key. The cursor will stop between the boxes to wait for input. Type in answers to the questions between boxes. If you make a mistake, the DELETE key erases text to the left. To move within what you have written, use arrow keys.

To print your essay, hold down CONTROL and PRESS P. To save your essay for future revisions, hold down CONTROL and PRESS S. Where prompted, type in the name of the file to be saved.

*** CAUTION!! To preserve original files, give each new file a name of its own (i.e., Draft 1, Draft 2, etc.). ***

USING COMPUTERS IN TEACHING
3

Data Analysis and Problem Solving

Subject Area: Social Studies
Grade Levels: 9-12

PARTICIPANT OBJECTIVES

As a result of participating in this teleconference and completing the related activities, the participant will:

1. Be aware of uses for data management software and specialized files in social studies classes.

2. Be aware of how the use of data management software and specialized files can contribute to the development of skills needed to be effective decision makers and problem solvers.

3. Be encouraged to create and use specialized files in their classes.

PRESENTATION SYNOPSIS

This teleconference focuses on the importance of using the computer in some of the most critical areas of the social studies curriculum--those that relate to gathering, organizing, analyzing, and using information for decision making, problem solving, and planning. The instructor will discuss the importance of using data management software and specialized files as the participants proceed through the session, which will include:

- Review of social studies objectives that are the focus of the projects that are part of this lesson and found in Appendix 3.

- Study of the contents of the data files.

- Sample searches and sorts.

- Demonstration of how a hard copy of the data can be produced for use in solving a problem or making a decision.

- Discussion of how the data may be used by students to complete one of the projects in Appendix 3.

- Explanation of the context for appropriate use of such an activity.

- Discussion of long-term uses for data files in the social studies.

TELECONFERENCE PREPARATION

1. Read the material in Appendix 3.

2. Identify places in the social studies curriculum where the type of information found in Appendix 3 on social and economic conditions in Latin America and Africa can be used.

3. Practice loading the program and sample data file, "World."

4. Practice using the file by answering a few of the questions in Project 1.

Study Questions

1. What are the differences in limited-use and multi-use data bases?

2. What are at least four vocabulary words, or terms, related to data bases?

TELECONFERENCE FOLLOW-UP

Discussion Topics:

- Sources of information that can be used to create new files.

- Management of computer use in a typical class.

- Potential uses of the file used in the demonstration and the type of files that are needed.

INDEPENDENT PRACTICE/PERSONAL GROWTH ACTIVITIES

1. Work through the three projects in the Student Activities found in Appendix 3.

2. Create a new project to be solved using the sample file.

3. Create a new file and new set of projects.

4. Engage your students in these activities if you find them appropriate to your courses and classes.

REFERENCES/RESOURCES

Data Pursuits: Countries of the World (InfoMaster data file), Grolier, 1985.

"Get Organized for Good With Data-base Software", Steve Morgenstern, Family Computing, March, 1986, pp. 30-35.

"Keeping Track" (videocassette), The Production Group, Inc., 1985.

MECC Dataquest: The Presidents, MECC, 1985.

NewsWorks (Appleworks data file), Newsweek Education Department, 1985.

Scholastic pfs: Curriculum Data Bases (PFS data file), Scholastic, 1985.

Teacher Handbook - Social Studies K-12, North Carolina Competency-Based Curriculum, Department of Public Instruction, Raleigh, 1985.

Toward a Better World Kits, World Bank Publications Sales Unit, 1818 H Street, N.W., Washington, D. C. 20433, 1981.

RELATED USES

The software used in this teleconference may be used to create files for a variety of topics in all subject areas.

USING COMPUTERS IN TEACHING 4

Data Collection, Analysis, and Graphing

Subject Area: Science
Grade Levels: 9-12

PARTICIPANT OBJECTIVES

As a result of participating in the teleconference and completing the related activities, the participant will:

1. Acquire awareness and understanding of how to use computers for data collection, analysis, and graphic representation in the science instructional program.

2. Understand the philosophy and methodology for the North Carolina science curriculum.

3. Understand the importance of teaching science process skills and how computer interfacing programs can assist.

4. Become aware of data analysis software that can assist teachers and students with experimental data.

5. Become aware of software programs to aid student understanding of graphic representations.

6. Become aware of science and computer competency goals in the Teacher Handbook - Science K-12.

PRESENTATION SYNOPSIS

This program will briefly discuss the philosophy and methodology for the new Standard Course of Study and the Teacher Handbook for the competency-based curriculum. The role of computers in the new science curriculum will be examined, and examples of software programs will be demonstrated. Software demonstrations will support the need for more active involvement in laboratory and experimental science.

The software demonstration will consist of three programs. They are:

1. An interfacing program using temperature and light probes to collect experimental data.

2. A data analysis program will illustrate an easy-to-use method of data management and analysis.

3. A graphics program will illustrate methods of improving student graphing and integrated science process skills.

TELECONFERENCE PREPARATION

1. Read the science philosophy in the Standard Course of Study, pages 255-262.

2. Review examples of integrated science process skills in Introductory Science Skills.

3. Review the manual for the SCIENCE TOOLKIT and be familiar with the software.

Study Questions:

1. How do you now use computers to enhance your teaching?

2. What can the computer do for the science teacher?

3. What is the major goal of eaucation and how does science support this goal?

TELECONFERENCE FOLLOW-UP

Review Questions:

1. What problems do you foresee with using computers in science instruction?

2. What other ways do you think computers can assist the science instructional program? And other subject areas?

INDEPENDENT PRACTICE/PERSONAL GROWTH ACTIVITIES

1. Experiment with SCIENCE TOOLKIT, SNAPGRAPH, and RESEARCH ASSISTANT.

REFERENCES/RESOURCES

Gabel, Dorothy, Introductory Science Skills, 970 pages, Waveland Press, Prospect Heights, IL, 1984.

North Carolina Standard Course of Study, Department of Public Instruction, Raleigh, 1985.

Research Assistant, Celcor Inc., Raleigh, NC, 1985.

Science Toolkit User's Manual and Experiment Guide, Broderbund Software, Inc., 1985.

Snapgraph, Celcor Inc., 1985.

Teacher Handbook - Science K-12, North Carolina Competency-Based Curriculum, Department of Public Instruction, Raleigh, 1985.

RELATED USES

Skills discussed in this program integrate well with computer, mathematics, and communications skills in the *Teacher Handbook* for those subject areas in the competency-based curriculum.

USING COMPUTERS IN TEACHING
5

Arts Education

Subject Areas: Music & Visual Arts
Grade Levels: 6-12

PARTICIPANT OBJECTIVES

As a result of participating in the teleconference and completing the related activities, the participant will:

1. Be aware of how the computer can be used in music education.
2. Develop skill in the use of the computer for music instruction.
3. Develop skill in the use of the computer for testing and evaluation in music instruction.
4. Be aware that teachers can teach traditional art concepts with the computer using the Koala Pad.
5. Be aware that computer art is a new art form.

PRESENTATION SYNOPSIS

The first part of the teleconference focuses on the utilization of the computer and computer programs in developing the student's ability to hear music more precisely, as well as to respond with greater precision.

- MELODIC DICTATOR will be used to explore melodic understanding and to develop the ability to understand and identify melodic intervals.
- SEBASTIAN II will focus on the identification of errors in melodies which are presented by the computer.
- The SOUNDCHASER synthesizer will demonstrate the use of the computer to aid in synthesizing musical sounds, developing an understanding of the components of these sounds.
- Additional possibilities for use of the computer in public school music education programs will be discussed.

The second part of the teleconference will introduce the participants to the Koala Pad input device and the KOALAPAINTER software as tools for teaching ιraditional visual arts concepts. Computer art as a new art form will be discussed, and samples of students' art will be shown.

TELECONFERENCE PREPARATION

1. Read the attached reviews of THE MELODIAN DICTATOR and SEBASTIAN II.
2. Read the material that accompanies these two software packages.
3. Read the music and visual arts parts of the Standard Course of Study and Teacher Handbook - Arts Education K-12.
4. Read the Koala Pad manual and the material that accompanies the KOALAPAINTER. Follow instructions for connecting the Koala Pad to the computer and have it ready for use.
5. View "Micros in the Arts" from READY OR NOT, School Television series.

Study Questions

1. What are the potential uses of computers in music and visual arts education programs?
2. How can teachers use the computer to enable/encourage students to be creative?

TELECONFERENCE FOLLOW-UP

Review Questions

1. How does computer technology relate to the music and visual arts portions of the Standard Course of Study and the Teacher Handbook - Arts Education?
2. How can teachers go about getting enough computers and software into their classrooms for students to use as a basic part of their instruction?

INDEPENDENT PRACTICE/PERSONAL GROWTH ACTIVITIES

- Practice using three software packages demonstrated in the teleconference.

REFERENCES/RESOURCES

Electronic Art: Computers in the Art Room (brochure), Department of Public Instruction, Raleigh.

Electronic Instruction (brochure), Department of Public Instruction, Raleigh.

GRAPHICS EXHIBITOR for use with the Koala Pad touch tablet, Koala Technologies Corporation, Santa Clara, CA (necessary for printing with Imagewriter).

Koalapainter Owner's Manual, Koala Technologies.

Melodious Dictator, Temporal Acuity Products.

"Micros in the Arts," READY OR NOT, School Television series, Department of Public Instruction, Raleigh.

"North Carolina's Electronic Art: Computers in The Art/Classroom - 1985 Summary" (duplicated report), Department of Public Instruction, Raleigh.

North Carolina Standard Course of Study and Introduction to the Competency-Based Curriculum, Department of Public Instruction, Raleigh, 1985.

Sebastian II, Temporal Acuity Products.

Teacher Handbook - Arts Education K-12, North Carolina Competency-Based Curriculum, Department of Public Instruction, Raleigh, 1985.

USING COMPUTERS IN TEACHING
6

Authoring and Graphing

Subject Areas: Foreign Language, Grades 9-12
Mathematics, Grades 11-12

PARTICIPANT OBJECTIVES

As a result of participating in the teleconference and completing the related activities, the participant will, for Part 1 on authoring:

1. Understand what an authoring system is.

2. Be able to author a short lesson with the BRAINZ-GAMZ modified lesson-authoring system.

3. Be able to recognize the application of authoring systems to foreign languages and other subject areas.

For Part 2 on graphing:

1. Learn the mechanics of using a computer graphing program.

2. Explore the potential and limitations of using a computer to graph functions.

3. Use a simple graphing program on the Apple II to study trigonometric functions (with changes in amplitude, period, and phase shift), polynomials (shapes, vertices, roots), rational functions (asymptotes) and tangents lines.

PRESENTATION SYNOPSIS

Part 1: Authoring

An authoring system enables a teacher to develop courseware without having to use a programming language. Therefore, the learning period is relatively brief and the actual time needed to produce courseware is shortened. Authoring systems are particularly useful to foreign language teachers because they allow a teacher to individualize the content of a CAI lesson in accordance with the language background of particular students.

This presentation focuses on BRAINZ-GAMZ, a modified lesson-authoring system which allows teachers to enter their own material in five game and one test format. In addition, teachers can store questions and answers of four types (multiple choice, true-false, fill in the blank, direct question) on a data base which can then be utilized in three game on the one test format. The program consists of three diskettes--the authoring diskette, the data base diskette, and the lesson diskette. Data base and lesson diskettes can be copied. The individual lessons can be created in English, Spanish, French, and German.

During this presentation, teachers will work at the computer and follow the procedure for authoring one game.

The presentation will conclude with an explanation of the data base portion of the program as well as the application of BRAINZ-GAMZ to all subject areas.

Part 2: Graphing

A short graphing program for the Apple II will be used to illustrate various instructional techniques. The program GRAPHIT and a few warm-up exercises will be made available to viewers in advance of the presentation so that participants should be prepared to cover the topics below:

1. Discussion of the difference between automatic and manual scaling of graphs.

2. Comparison the graphs of $y = f(x)$ and $y = Af(BX+C)$.

3. Location of the zeros of a function to within a given tolerance.

4. Discussion of the effects of vertical asymptotes on a computer graph.

5. Comparison of a curve to its tangent line.

6. Brief discussion of calculus applications.

TELECONFERENCE PREPARATION

Part 1: Authoring

1. Read the ERIC sheet on CAI and Second Languages in Appendix 4.

2. Read article, "Three Major Approaches to Developing Computer-Assisted Language Learning Materials for Microcomputers," in Appendix 4.

3. Play at least one game on the computer from supplied lesson diskette, BRAINZ-GAMZ.

Part 2: Graphing

1. Read Preparation section of the handout, "Graphing Functions with a Computer," in Appendix 4.

2. Complete the exercises in the "Preparation" section.

Study Questions

1. The Apple deals with discrete rather than continuous data. For example, points can be plotted only at 280 x 192 positions on the screen, and there is a smallest positive real number. What are the limitations of graphing a continuous function $y = f(x)$ on such a machine?

2. When is it appropriate to use a computer to graph functions rather than sketching them by hand? When is it inappropriate?

TELECONFERENCE FOLLOW-UP

Part 1: Authoring

Review Questions

1. Why is an authoring system useful?

2. How does it differ from word processing?

3. How can you use BRAINZ-GAMZ in your subject area?

Part 2: Graphing

Review Questions

1. What is the difference between manual and automatic scaling in the GRAPHIT program? When should each be used?

2. How do you go about putting more than one curve on the same graph?

3. Discuss the various things that can go wrong when graphing the function $y = 1/x$ on the Apple.

INDEPENDENT PRACTICE/PERSONAL GROWTH ACTIVITIES

Part 1:

1. Author a minimum of two different games; use the data base in at least one of them.

Part 2: Graphing

1. Read the Follow-up section of "Graphing Functions with a Computer."

2. Complete the follow-up exercises in the article.

REFERENCES/RESOURCES

Part 1: Authoring

<u>Brainz-Gamz</u>, Bainum Dunbar, Inc., 1985.

David H. Wyatt, "Three Major Approaches to Developing Computer-Assisted Language Learning Materials for Microcomputers," <u>CALICO Journal</u>, Vol. I, No. 2, September, 1983.

Minibibliography Sheet - "Computer-Assisted Instruction and Second Languages," ERIC Clearinghouse on Languages and Linguistics.

Part 2: Graphing

<u>Graphit</u> (diskette), Mark Harris, released in the public domain, 1986.

"Graphing Functions with a Computer" (duplicated handout), Harris, 1986.

The following software package for the Apple II includes a 200 page manual and a variety of graphing programs for more extensive follow-up:

Brown & Harris, ARBPLOT, CONDUIT/Harper & Row, 1982.

APPENDIX

1

FINAL DRAFT

Computer Awareness Competency Goals, Objectives, and Measures from the Teacher Handbook for

Library/Media & Computer Skills

K-12

NORTH CAROLINA
COMPETENCY-BASED
CURRICULUM

Division of School Media Programs
Division of Computer Services

Media & Technology Services
Department of Public Instruction
Raleigh, North Carolina 27611

NORTH CAROLINA COMPETENCY-BASED CURRICULUM

LIBRARY/MEDIA AND COMPUTER SKILLS

GRADES K-12

Division of School Media Programs
Division of Computer Services
Media and Technology Services
in cooperation with
Instructional Services
North Carolina Department of Public Instruction

Raleigh, North Carolina
1985

LIBRARY/MEDIA AND COMPUTER SKILLS

Grade Level: K Skills/Subject Area: Computer Awareness

COMPETENCY GOAL 6: The learner will demonstrate an understanding of computers, their operation, and their possible application to solving relevant problems.

Objectives	Measures
6.1 Identify the physical components of computer hardware and software (e.g., monitor, keyboard, disk drive, diskette).	6.1.1 Name the computer component when shown its picture. 6.1.2 Point to or touch each component when directed.
6.2 Identify the computer as a machine or tool that helps people play and work.	6.2.1 Choose the computer as an object which is not alive, when given a variety of pictures showing objects which are alive and which are not. 6.2.2 Choose the picture of a computer as an example of a machine, when given a variety of pictures of plants, animals, and machines.
6.3 Follow simple oral, graphic, and pantomimed directions for using a computer.	6.3.1 Locate specific keys on the keyboard and move fingers to those keys when directed. (May use model or actual keyboard.) 6.3.2 Follow directions from an audiotape that indicate certain keys be pressed on the computer (e.g., "Put your finger on the 'A' key and press it lightly", or "Touch the spacebar lightly").

LIBRARY/MEDIA AND COMPUTER SKILLS

Grade Level: 1 Skills/Subject Area: Computer Awareness

COMPETENCY GOAL 6: The learner will demonstrate an understanding of computers, their operation, and their possible application to solving relevant problems.

Objectives		Measures	
6.1	Identify and demonstrate a working understanding of the simple computer terms load and run after being introduced to them.	6.1.1	Verbally describe the difference in the procedures for loading and for running a program.
		6.1.2	Arrange pictures to show the correct sequence for loading and running a program.
		6.1.3	Demonstrate the proper technique for loading and effectively running a program.
6.2	Describe what a computer can and cannot do for humans.	6.2.1	After being shown pictures of familiar activities, indicate which can be done by a computer.
		6.2.2	Draw a picture that shows something a computer can or cannot do.
6.3	Describe the acceptable procedure for software handling.	6.3.1	Use a model to show the correct way to handle software. (Model may be cardboard or discarded software.)
		6.3.2	From a group of pictures that show correct and incorrect software-handling activities, choose those that show correct procedures.

LIBRARY/MEDIA AND COMPUTER SKILLS

Grade Level: 2

Skills/Subject Area: Computer Awareness

COMPETENCY GOAL 6: The learner will demonstrate an understanding of computers, their operation, and their possible application to solving relevant problems.

Objectives	Measures
6.1 Identify and define simple computer terms after being introduced to them (e.g., run, program, list, enter, load).	6.1.1 Use the computer word or a definition of the word to find the word in a word search puzzle.
	6.1.2 After hearing the definition for a computer term, select the correct term card from a group of flash cards.
6.2 Identify several roles the computer plays in daily lives.	6.2.1 After class discussion, tell how computers can help community helpers do their jobs (e.g., policeman, grocer, school principal).
	6.2.2 Find examples of computerized information in the home. Contribute these to a bulletin board display (e.g., grocery tape, bills, punched cards, labels with Universal Product Codes).
6.3 Demonstrate an understanding of software as a set of instructions, called a program, that tells a computer what to do. Continue to demonstrate proficiency in the use of software.	6.3.1 Act out a familiar task, such as how to feed a pet, with needed steps in the correct order.
	6.3.2 Record directions on an audiotape that tell another student how to do something. Share the recording and have the task correctly performed (e.g., Take paper shapes from an envelope and arrange them in a certain order).
	6.3.3 Given a written list of steps necessary to perform a task that includes one unnecessary step, indicate the step that is not needed.
	6.3.4 Demonstrate greater independence in using software by loading and running a program without teacher supervision.

Grade Level: 3

Skills/Subject Area: Enrichment

COMPETENCY GOAL 5: The learner will expand reading, listening, and viewing interests by using a variety of media for personal growth, vocational pursuits, and recreation.

Objectives	Measures
5.7 Understand and acknowledge the concept of authorship (writing and illustrating).	5.7.1 Give credit to the author when presenting a simple report.
	5.7.2 Cite the sources of information when writing a simple report, e.g, Let's Discover, Vol. 3, p. 7.
5.8 Understand the rights of others in the media center.	5.8.1 Write and present a skit with classmates that shows proper media center behavior.
	5.8.2 Follow directions given by media personnel for using different areas of the media center, e.g., listening/viewing, informal reading, reference.
5.9 Share knowledge about media center use with others.	5.9.1 Write a short article for the school newspaper that tells about a recent media center activity.
	5.9.2 Write a letter home or to grandparents that describes an activity planned for Children's Book Week or National Library Week.
5.10 Treat the facility, materials, and equipment with care and show respect for media center personnel when using the media center independently or in groups.	5.10.1 Demonstrate to younger students the proper care of media materials.
	5.10.2 Show, by example, how to be a good media center user, e.g., following directions for using audiovisual equipment, using quiet voices, returning materials to the proper location.

Grade Level: 3 Skills/Subject Area: Computer Awareness

COMPETENCY GOAL 6: The learner will demonstrate an understanding of computers, their operation, and their possible application to solving relevant problems.

Objectives	Measures
6.1 Identify and define in simple operational terms the names for three main computer parts, i.e., input, processor, and output.	6.1.1 Match two groups of flash cards--one labeled computer part terms and the other with appropriate definitions. (Example--processor/the part of a computer that receives, works with, and sends out numbers and words)
	6.1.2 After class discussion, locate pictures of computer hardware. Classify them according to input, processor, or output and create an individual or group display.
6.2 Know the order in which information passes through the three main computer parts, i.e., input, processor, output.	6.2.1 In the correct processing order, step on each of three large pieces of paper labeled with input, processor, and output.
	6.2.2 Given an example of an input and associated processing rule or object, name the output. (Directions: Label a box with a processing rule or object. Give the appropriate input and have the student tell the output, e.g., The box is labeled toaster; the input is bread; the output is toast. The box is labeled addition; the input is two and two; the output is four.)

LIBRARY/MEDIA AND COMPUTER SKILLS

Grade Level: 4 Skills/Subject Area: Computer Awareness

COMPETENCY GOAL 6: The learner will demonstrate an understanding of computers, their operation, and their possible application to solving relevant problems.

Objectives	Measures
6.1 Name the parts of a computer (input, memory, central processing unit, arithmetic unit, and output) and give a simple description of how each works.	6.1.1 Label each part of a diagram that shows the five parts of a computer.
	6.1.2 Complete a crossword puzzle of computer terms which emphasizes the five parts of a computer.
	6.1.3 On a written test, match the five parts of a computer with the correct definition or simple description of how the part works.
6.2 Identify examples of objects in the home that contain a computer or computerized parts.	6.2.1 Cut pictures of computer objects in the home from magazines and contribute them to a class display or create an individual scrapbook.
	6.2.2 Write a story that tells how a computer helps with some task at home (e.g., a Computer Helps Prepare Our Meals; Computers In My Toys).
6.3 Save information from the computer on a storage device. Retrieve information from a storage device to a computer. (Storage device may be a diskette or a cassette. Information may be teacher-developed, student-modified, or a simple student-developed program.)	6.3.1 Write on paper the correct sequence of commands necessary to save and retrieve text.
	6.3.2 Demonstrate with a formatted diskette how to save text on the diskette and how to retrieve the same text.

LIBRARY/MEDIA AND COMPUTER SKILLS

Grade Level: 5 Skills/Subject Area: Computer Awareness

COMPETENCY GOAL 6: The learner will demonstrate an understanding of computers, their operation, and their possible application to solving relevant problems.

Objectives	Measures
6.1 Identify and define the basic vocabulary necessary for word processing (e.g., edit, save, print, delete).	6.1.1 Create a word processing dictionary that includes terms, definitions, and appropriate graphics.
	6.1.2 Match, on a written test, word processing terms and/or commands to their correct definition or function, e.g., edit/change text.
6.2 Explain the role of the computer as word processing tool in society.	6.2.1 Identify five personal or business activities that can use word processing programs.
	6.2.2 After a class discussion of a word processing program (e.g., what it is, how it works, how it can help in all kinds of writing), develop a commercial or advertisement that promotes the advantages of word processing to a target audience.
6.3 Use a word processing program for beginning level activities.	6.3.1 Use a sample document and a word processing program to input and edit (delete, add, replace) text.
	6.3.2 Save and retrieve the text using sample document files.
	6.3.3 Use a word processing program to produce a simple, printed document.

LIBRARY/MEDIA AND COMPUTER SKILLS

Grade Level: 6 Skills/Subject Area: Computer Awareness

COMPETENCY GOAL 6: The learner will demonstrate an understanding of computers, their operation, and their possible application to solving relevant problems.

Objectives	Measures
6.1 Identify the common computer languages (BASIC, Logo, COBOL, Pascal, Fortran) and their typical application in society.	6.1.1 Given a list of common computer languages, match the language and the typical application of that language (e.g., BASIC with education, COBOL with business, Fortran with science/mathematics).
	6.1.2 Give an oral, visual, or written presentation about the development and use of a particular language.
6.2 Demonstrate a knowledge of the major events, people, and devices in the history of computing.	6.2.1 Match a picture of a historical computing device with the appropriate name.
	6.2.2 On a timeline, label significant inventions that influenced the history of computers and computing. Show the span of time for the four generations, e.g., the vacuum tube, the transistor, the integrated circuit, the chip.
	6.2.3 Give an oral, visual, or written presentation on some aspect of the history of computers (e.g., Pascal, Hollerith, abacus, microchip).
	6.2.4 Create a chart or graph relating how the size and processing speed of computing machinery has changed through the different generations.
6.3 Demonstrate an increasing proficiency in the use of a word processing program.	6.3.1 Use a sample document and a word processing program to practice previously introduced editing techniques and the use of more advanced editing features (e.g., moving a block of text).
	6.3.2 Use a word processing program to create and produce a printed document such as a short letter, poem, or a report.

LIBRARY/MEDIA AND COMPUTER SKILLS

Grade Level: 7

Skills/Subject Area: Computer Awareness

COMPETENCY GOAL 6: The learner will demonstrate an understanding of computers, their operation, and their possible application to solving relevant problems.

Objectives	Measures
6.1 Explain how the size, cost, and use of a microcomputer differ from that of a minicomputer and a mainframe computer.	6.1.1 Given a chart of the general features of the three types of computers, verbally compare the major features (e.g., size, cost, memory, operations per second).
	6.1.2 On a written test, match the type of computer with the appropriate application (e.g., mainframe/census data).
	6.1.3 Use a collection of catalogs and computer advertisements to select and label an example of a mainframe, mini, or microcomputer and explain the selection.
6.2 Understand the uses of a computer to process data.	6.2.1 Given a noncomputerized data base (e.g., encyclopedia, telephone directory, or school's card catalog), explain how the data is stored and retrieved.
	6.2.2 Give an oral, written, or visual presentation on the advantages (speed, accuracy, quantity) resulting from using the computer to process data; relate these advantages to school (report cards, equipment inventory) and/or everyday use (police checks of license plate numbers).
	6.2.3 After studying a computerized data base (e.g., View a film on the subject, take a field trip to a facility using a computerized data base, or interview someone using the computer for the data base), prepare a bulletin board or report on the advantages of using the computer for the data base function researched.

LIBRARY/MEDIA AND COMPUTER SKILLS

Grade Level: 8

Skills/Subject Area: Computer Awareness

COMPETENCY GOAL 6: The learner will demonstrate an understanding of computers, their operation, and their possible application to solving relevant problems.

Objectives	Measures
6.1 Explain the impact of computers on public and private lives and the laws protecting privacy.	6.1.1 Given a list of computer activities (e.g., national security, banking), state why data privacy is important in each of these areas.
	6.1.2 List at least five data banks that contain personal information (e.g., credit bureau, driver's license, insurance, bank, doctor's office) and describe the purpose of each data bank.
	6.1.3 Given a specific topic (e.g., voting, census taking, retailing), write a paragraph or organize a skit about the impact of computers in that area in the present or in the future.
	6.1.4 Given a computer piracy example (unauthorized access of student records), write an editorial or design a poster supporting laws or legislation to protect personal privacy.
6.2 Demonstrate an increasing proficiency in the use of a data base program.	6.2.1 Given a particular topic and a predesigned input screen, collect and input data into the data base (e.g., TOPIC: North Carolina counties--CATEGORIES: name, population, geographical features).
	6.2.2 Make a back-up copy (diskette/cassette) of the data file containing the input data.
	6.2.3 Select, retrieve, and analyze data from the data file to create a visual display and/or to answer specific comparison questions (e.g., Create a bar graph that compares population data).

LIBRARY/MEDIA AND COMPUTER SKILLS

Grade Level: 9　　Skills/Subject Area: Computer Awareness

COMPETENCY GOAL 6: The learner will demonstrate an understanding of computers, their operation, and their possible application to solving relevant problems.

Objectives	Measures
6.1 Demonstrate an understanding of computer copyright and software piracy issues.	6.1.1 Write an editorial or design a poster supporting the computer copyright laws.
	6.1.2 Participate in a round-table discussion or debate on software piracy, ownership rights, and user responsibilities.
6.2 Given the opportunity, select appropriate software for class and personal use.	6.2.1 Given a list of class assignments and a collection of computer software, select and describe the appropriate use of the software for that assignment.
	6.2.2 Given the generic name for a piece of software (e.g., data base, word processing, spreadsheet, CAI), cite several uses for the software and how it could be used to solve the relevant problems of a student.
	6.2.3 Design a promotional poster to sell a piece of software for a specific personal/home use.

LIBRARY/MEDIA AND COMPUTER SKILLS

Grade Level: 10 Skills/Subject Area: Computer Awareness

COMPETENCY GOAL 6: The learner will demonstrate an understanding of computers, their operation, and their possible application to solving relevant problems.

Objectives	Measures
6.1 Identify examples and analyze the societal impact of advanced computer technology, e.g., robotics, voice emulation,artificial intelligence, interactive video.	6.1.1 Collect articles from magazines, newspapers, and catalogs on advanced computer technology and organize by types of technology for a class display or individual current events report. Report or contribute to Information File in the media center.
	6.1.2 Prepare an editorial newscast to report on the contents of selected articles about advanced computer technology.

LIBRARY/MEDIA and COMPUTER SKILLS

Grade Level: 11 Skills/Subject Area: Computer Awareness

COMPETENCY GOAL 6: The learner will demonstrate an understanding of computers, their operation, and their possible application to solving relevant problems.

Objectives	Measures
6.1 Explain the appropriate use of and the procedure for accessing on-line reference services.	6.1.1 State on a written test what basic computer hardware and software is needed to access an on-line reference service.
	6.1.2 Given a description of various data bases of an on-line reference service, select the specific data base to use to retrieve information on a stated topic in order to share in a class discussion or on a written test.
	6.1.3 Develop a chart for the procedural steps to access on-line reference services.

LIBRARY/MEDIA AND COMPUTER SKILLS

Grade Level: 12 Skills/Subject Area: Computer Awareness

COMPETENCY GOAL 6: The learner will demonstrate an understanding of computers, their operation, and their possible application to solving relevant problems.

Objectives	Measures
6.1 Define, identify and analyze emerging telecommunications technology.	6.1.1 On a written test, define the term <u>telecommunications</u> and from a list, select and describe at least two examples of telecommunications technology (e.g., laser disk, modem, dish, satellite).
	6.1.2 Given the name of a specific telecommunications technology, present an oral, visual, or written report about the technology, i.e., its development, applications, cost, and impact.
	6.1.3 After a class discussion of emerging technology, cite a local, county, state, or national need that could be aided by telecommunications and use information from current resources or on-line services to suggest how one of these technologies could address this need.

GOAL 6: Computer Awareness

K	1	2	3	4	5	6	7	8	9	10	11	12
COMPUTER VOCABULARY 1.PHYSICAL COMPONENTS OF HARDWARE AND SOFTWARE - MONITOR, KEYBOARD, DISK DRIVE, DISKETTE ...	→	→	→	→	→	→						
				1.-INPUT, MEMORY, CENTRAL PROCESSING UNIT, ARITHMETIC UNIT, OUTPUT								
	1.COMPUTER OPERATIONAL TERMS -LOAD, RUN	→	→	→	→	→						
		1.-PROGRAM, LIST, ENTER, ...	1.-INPUT, PROCESSOR, OUTPUT		1.-WORD PROCESSING TERMS. EDIT, SAVE, PRINT, DELETE, ...	1.BASIC, LOGO, COBOL, PASCAL, FORTRAN						
COMPUTER ROLE 2.PROCESSING TOOL	→	→	→	→	→	→						
	2.-LIMITATIONS											
		2.SOCIETAL APPLICATIONS	→	→	→	→	→	→	→	→	→	→
				2.-CONTRIBUTIONS TO HOME ENVIRONMENT	2.-CONTRIBUTIONS TO WORLD OF WORK	2.-HISTORICAL DEVELOPMENT	1.-THREE TYPES OF COMPUTERS	1.-SOCIETAL IMPACT	1.-COPYRIGHT AND PIRACY ISSUES	1.-ADVANCED COMPUTER TECHNOLOGY		1.-TELECOMMUNICATIONS TECHNOLOGY
COMPUTER USE 3.FOLLOWING INSTRUCTIONS	→	→	→	→	→	→	→	→				
	3.COMPUTER SOFTWARE -HANDLING	→	→	→	→	→	→	→	→			
		3.-PROGRAM DESCRIPTION							2.-PROGRAM SELECTION			
			2.PROCESSING ORDER	→	→	→	→	→				
				3.-SAVING, RETRIEVING								
					3.SPECIAL APPLICATIONS -WORD PROCESSING	➤ 3.	→	→	→	→		
											1.-ONLINE REFERENCE SERVICES	
							2.-DATA BASES	➤ 2.				

APPENDIX

2

A Dozen Ways for English Teachers to Use Microcomputers

R. Baird Shuman

Few English teachers nowadays question whether or not they should use microcomputers in their instruction. These machines are beginning to appear in even the remotest schools, and many school districts are encouraging their teachers to buy their own personal microcomputers by offering machines to them at drastically reduced prices and by offering them no-interest loans to be paid back through payroll deductions so that they can have microcomputers in their homes.

However, the prevalence of microcomputers does not, in itself, make teachers any less apprehensive about using these machines, and only when they overcome these apprehensions will the jittery make the fullest use of the exciting new technology available to them. Many teachers, before they take the first step into using microcomputers and word processors, have little idea what they will be able to do with the machines. Only when they develop a clear notion of the uses to which this new equipment can be put and become convinced that using these machines is within their capability will they start to develop the confidence to make microcomputers an integral part of their normal teaching routine.

Those who are just beginning to test the waters may find it helpful to identify a dozen or so specific uses to which they can put their microcomputers fairly immediately without a great deal of computer training. The following suggested uses do not represent anything like a comprehensive list but rather indicate some quite easy initial uses of the machines. Some of the uses suggested serve essentially administrative or housekeeping functions. Others serve specifically instructional ends. Yet others fall between these two functions.

Administrative Uses

1. *Keeping class records, including attendance and grades.* Record keeping often takes more teacher time than it should, but records must be kept. The microcomputer will keep them on floppy disks and will perform such tasks as indicating patterns of absences (after holidays, before or after major tests, before or after weekends, etc.) which might suggest to teachers ways of reducing absences. Averaging grades becomes virtually automatic when the microcomputer is used, and comparative grading from student to student or section to section is made easy.

2. *Keeping the whole of a student's class work on a disk.* This makes it easy to check student progress through the semester and to identify trends in the types of errors students make. Error analysis is probably the most important step in showing teachers precisely how to proceed in individualizing instruction for students, and such error analysis, which can extend over a semester or over the whole course of a student's high school career, will be available to teachers at the typing in of a simple command.

3. *Keeping track of student achievement.* This will reveal areas in which a whole class needs additional work and will also point to the areas in which just one or two students need additional instruction. The microcomputer, of course, can give the specific, individualized help that students need without wasting the time of the entire group. Also, learning

of skills will be accelerated when the computer serves the function of tutor because the learning will be sequential, each step will be logically reinforced, and the computer will not embarrass students who are slow to learn but will instead, with its infinite patience, move students bit by bit from where they are to where they need to be in their understanding of whatever is under consideration.

4. *Checking sentence length and variety.* Microcomputers can be programmed to give the Fog Index of a given piece of writing or to review the piece in such a way that readability indexes like Fry's can be assigned to the composition being considered. This check can give students excellent clues to their own writing styles and can indicate to teachers the salient patterns in the assorted writing styles of a class of 30 or 35 students. Microcomputers can also be programmed to show variety of sentence length and, if the punctuation is proper, to give some indication of sentence complexity by counting punctuation marks.

Combined Administrative and Instructional Uses

5. *Storing information for students to use in their writing.* The microcomputer can be used as a notebook of sorts. Information can be fed into it and held for future use. Students writing research papers will find this use extremely helpful. Also, students can feed the microcomputer impressions and store in it the kinds of information usually found in their journals. This information can be summoned forth easily as soon as students have need of it.

6. *Tutoring students in routine drill work.* The microcomputer is invaluable in helping students step by step through tedious drills in usage, mechanics of expression, and punctuation. Valuable software exists in these areas, and it is programmed to relieve students of the embarrassment of not learning something as quickly as their peers. The microcomputer will go over material again and again adapting the level of difficulty to the individual. The microcomputer constantly checks and patiently adjusts the level of difficulty to whatever the learner can handle. The transaction between the computer and the learner is private, and the only competition students feel is with themselves as they are measured against their past performance. Such a learning atmosphere is ideal for students who are slow or who lack self-confidence.

7. *Composing and varying objective tests.* Teachers can compose objective tests quite easily on microcomputers. They can format tests so that once the first question is set up, similar questions are formatted in the same way. Teachers can easily alter the order in which questions appear so that in crowded testing conditions, cheating is minimized. The microcomputer makes it easy to run error-analysis of test items from which the instructor can begin to see emerging patterns. This information can be valuable in suggesting what work needs to receive more attention and review during future classes.

8. *Checking for spelling and typographical errors.* Dictionaries of 40–50,000 words are now available for nearly all microcomputers, and these programs will check every word in a written piece against their wordlists. One may add words to the wordlist the program provides as individual needs arise. One drawback of the wordlist is that it will indicate that most proper names are not in the list, so if some proper names are likely to recur often in a piece of writing, it would be well to add these names to the wordlist through a quite simple operation. It may also, in some situations, be necessary to add specific terminology, such as chemical or biological terms, to the wordlist so that they will not appear on the screen as possible misspellings. When one uses the microcomputer's dictionary, it is necessary to accept or reject each questionable spelling that is identified, and this can be time-consuming in some specialized kinds of writing. Present dictionaries cannot distinguish between some pairs of homonyms like *there/their* or *its/it's.* The dictionary checks only spelling, not usage.

9. *Playing video games.* Teachers have long been hard put to know what to do with students who finish their work early and have done it well. If students are assigned more work, they soon realize it does not pay to finish early, so they practice what, in the days of slavery, was called the "slowdown." This ultimately leads to boredom and may have an adverse effect upon their overall performance in class. Now, if students finish their work early, there is no reason that they should not be permitted to play video games. Accommodating student interest in this way can do a great deal to motivate students to work efficiently, rapidly, and well.

Instructional Uses

10. *Writing and revising in class.* The most important function of the microcomputer for English teachers is surely its word processing function. Word processing enables students to write and revise in the classroom, and it adds considerable flexibility to the writing act. Many students, especially those with bad penmanship, can compose more productively on the keyboard than they ever have been able to do with pen and paper. Also, the finished result is much easier for teachers and classmates to read than are handwritten papers. Teachers have noticed that students with fluency problems often overcome them quickly and substantially when they get used to writing with the word processor. Revision becomes an adventure rather than a chore, as whole paragraphs are moved from place to place within a paper. The word processor will not turn a bad writer into a Nobel prize laureate in literature, but it will remove some of the stumbling blocks that students often encounter when they write.

11. *Formatting writing.* How a piece of writing looks on the page has a great deal to do with one's enthusiasm for reading it. The word processor can show how a piece looks single-spaced or double-spaced, set up with flush margins or with indented ones, set up with blocked material presented in a variety of ways, and other such visual matters. The microcomputer is especially good at formatting poetry, sometimes in most unusual ways. Concrete poetry is especially appropriate to word processing through the microcomputer. Playing with format is great fun and the microcomputer offers an infinite range of possibilities for experimental formats.

    Programs also exist that enable students to compose a poem and then to have it printed out with every possible combination of lines available from the lines given. This procedure often helps students to reformulate some of their work. In some arrangements, certain lines of poetry obviously will not work, and these combinations can be discarded. However, the microcomputer is likely to produce a number of workable combinations, and students can decide from these how their poem best communicates.

12. *Adapting tests to the student.* In the October 1983 *Phi Delta Kappan*, Wallace Judd wrote,

    > In computerized adaptive testing, . . . the computer selects items for the test while the test is in progress. If a student answers the first item correctly, the next item is harder. If a student gets the first item wrong, the next item is slightly easier. If the student misses that item, the computer selects a still easier item until the student answers one correctly. In this way, the test is soon composed of items that are at the right level of difficulty.

    Judd defends adaptive testing as a more humane approach to testing than is that of giving tests that bore the bright and frustrate the not-so-bright student. In adaptive testing, clearly the test becomes a major part of an ongoing learning process. Such cannot be said for much of the haphazard objective testing to which students nowadays are often subjected.

Afterword

This list is a mere beginning, but it may serve teachers and administrators in suggesting to them ways they can integrate the microcomputer into their schools. Teachers might wish to keep a notebook in which they jot down other uses they have found for the microcomputer as a learning tool. It is easy to imagine that such a notebook would fill up quickly because every time teachers and students use microcomputers, new ways of using them productively suggest themselves.

R. Baird Shuman teaches at the University of Illinois.

Computers and composing: The pros and cons

by Elaine McNally Jarchow

Some composition teachers are enthusiastically embracing the microcomputer as the writer's best friend, while others refuse to even accept computer compositions. Because I teach instructional computing courses and workshops to language arts teachers, I am convinced that the positive aspects of computer composing far outweigh the negative aspects. This comparison, though, discusses the positive results of student compositions produced with the help of the computer as well as the negative aspects.

10 Positive Conclusions

1. Writers can easily revise, edit and expand computer compositions.

A student who had just begun to use a word processor wrote, "There is something about knowing I can move paragraphs and delete whole sentences with relative ease that allows me to jump in and get started."

2. Computer writers are motivated to spend more time composing.

Students at a West Des Moines junior high school stand in line to use the Bank Street Writer.

3. Spelling checkers, thesaurus programs and electronic editors enhance student compositions.

Experienced writers like programs which can quickly highlight mechanical problems. This feature frees the writer to concentrate on logic, organization and style.

4. Computer composing can positively alter style.

Because the writer need not worry about spelling, punctuation and the overuse of certain words, s/he can begin to develop unique, creative responses to ideas.

5. Computer writers learn more about sentence combining and arranging content.

A student writer reported that, "You don't just work with a flat piece of paper. Being able to move words and passages around, keeps you interested."

6. Easily produced multiple drafts contribute to writer collaboration and peer review.

Shared drafts of computer compositions encourage critical analyses of successive drafts and peer review.

7. Computer writers are less defensive about taking suggestions.

8. Computer writers do not suffer from the blank page syndrome.

A student writer made the observation, "It has helped in getting my initial ideas on the paper instead of rehashing them over and over in my head and has cured my hang-up about spelling correctly in my first draft."

9. Clean drafts enable writers to revise more objectively and printed drafts eliminate bias against poor handwriting.

Writers enjoy the process of manipulating text and of evaluating clean drafts. Evaluators can concentrate on style rather than neatness.

10. Printed drafts boost writer self-confidence.

A student writer concluded, "Somehow I cannot judge the quality of my writing when it is full of crossouts and arrows. The computer has solved this problem by allowing me to obtain neat, revised drafts quite easily. . . . "

10 Negative Conclusions

1. Computer assisted writing denies some writers' need to physically manipulate text.

Some writers actually like to watch pen and hand move across the page, while others enjoy the act of cutting and pasting their text.

2. The uniqueness of each writers' composing process may be lost.

Highly specific word processors and electronic editors may result in an overly mechanized process as prewriting and rewriting become too structured.

3. The elecronic editor may not define style as the teacher or writer would.

Because the content analysis capability of the electronic editor is limited, a specific, narrowly defined concept of style may be internalized by the writer.

4. Computer composing can negatively alter style.

Many electronic editors discourage highly complex sentences. Electronic composing may produce many Hemingways and no Faulkners.

5. Electronic dictionaries and spelling checkers cannot detect some semantic errors.

Very few editors can help the writer distinguish between affect and effect or there and their. These common errors are often problems for many writers.

6. Electronic text can be lost.

Any writer who has labored long over a piece of electronic text only to see it erased with a mistaken stroke of a key can attest to this negative aspect.

7. Spelling checkers are time consuming.

Some extensive checkers analyze a six page paper very slowly. Writers may have to wait some time before revising their drafts.

8. Word processors are expensive.

9. Poor typing habits may be learned by writers.

Secondary teachers often lament that very young computer writers will be unable to learn proper keyboarding techniques in high school.

10. Writers cannot take word processors everywhere they go.

Until the microprocessor becomes more portable, electronic writing is limited to certain environments. Creative spurts may also be limited to these environments once the writer comes to rely on the computer as pen and paper.

Dr. Elaine McNally Jarchow is a Professor of Secondary Education at Iowa State University in Ames, Ia.

A New Contributor

Process and Processor: Is There Room for a Machine in the English Classroom?

Gail G. Womble

Little did I know what I would be getting myself and my students into when I agreed to try word processing in my tenth grade English classroom. Our district administration wanted to determine if the benefits of word processing justified equipping *English* classrooms with computers. When I was asked to field test, I agreed, but I had many concerns.

I had just completed a summer's training at the Northern Virginia Writing Project, and I worried that a machine might interfere and depersonalize writing. Further, I had always seen myself as a mechanical failure. Could I, who panicked when a movie projector acted up, cope with equipment even more expensive and complex? Also, how would I feel staying one short step ahead of my students as I learned, lesson by lesson, how to use the processor? Finally, and not to be underestimated, how would I solve the problem of integrating something new into an already crowded curriculum? I was starting a new team-taught American Studies course for accelerated students. The objectives listed in the Fairfax County *Program of Studies* and the requirements in literature, history, and humanities seemed endless. I ruefully recalled how glibly I'd dismissed this problem at a fall workshop. When asked by another teacher how I'd manage to fit the word processor into everything else that had to be done, I'd replied that it would simply be one of many ways I'd approach writing instruction. How much I had to learn..

I wondered if my students had their own concerns about using the word processor. So before we even saw our computer, I asked them to write in their journals what they felt about using a word processor for revising and editing their writing. I was amazed how frequently their concerns and hopes mirrored mine.

Kiki spoke for many of her classmates when she wrote, "I think I'd rather just write on paper; it's more personal." Amy added, "Writing is picking up a pen and paper and just writing. Not punching things from a keyboard onto a screen . . . ." Melissa was concerned that a machine would take the pleasure out of writing: ". . . half the fun of reworking a paper is scratching out and fixing words." I wondered if they were right. A skeptic and somewhat of a traditionalist myself, I suspected they were.

Many of my students were as awed by the equipment as I. They, too, were afraid. Debbie felt that ". . . just learning *how* to use it will be very difficult and confusing." Erin reminded me of myself when she wrote, "I'm not great with machines either." Pranee echoed the concerns of others when she thought about typing at a keyboard, "The thing that is really going to be a problem for me is typing. It may take longer for me to type than to write—how awful."

Of all the students' comments, though, Katy's touched me the most. "A machine is going to teach me how to write. I wish it luck. In all my years of school none of the teachers have been able to teach me how to write. At this moment I can't write fast enough to capture a thought before it's gone. I won't get anything down on the computer before the idea is gone." Katy's journal spoke to me as no other had. I didn't want the computer to be a threat to students. I didn't want to force it upon them. I decided to make learning how to use the equipment mandatory but not to insist that it become a primary part of every student's writing process.

I soon realized that even if I wanted every student to use the processor extensively that it would be impossible. I just didn't have enough computers. One of the most difficult problems the class and I had to solve was scheduling 107 students onto the one computer I did have. Debbie voiced the worry we all shared. "I just don't understand how we are going to work out a system so that everyone will get a chance at doing it. It seems like there will be mass confusion." But together we worked out the best schedule we could, using lunch time and an after-school period as well as our class time. The students' willingness to work on their own time was gratifying. I had few complaints and, on the whole, fewer noncompliants than I had expected. Some students used the word processor more than others, some hardly at all, but for the most part, we all pitched in and worked together to schedule as much hands-on time as possible.

I don't want to imply that all of our problems were solved by determining the schedule. I had difficulties I'd never anticipated, starting with an equipment breakdown on the very first day I introduced the class to the computer. The disk drive failed to operate correctly, and I had no idea why. I felt foolish standing in front of the class with the equipment not working. I had no idea what to do next. Everyone was very supportive, but some students were also concerned. If *I* had problems, they told me, how could *they* ever handle things? Mostly, though, what we all needed was familiarity with the equipment. After a few weeks, the comments from students echoed Barrie. "I think I understand how it works a little more, and it doesn't seem so hard anymore." I was glad we were all starting to feel more secure, because there were bad days—days when wheels fell off the computer cart, when the old television set we used as a monitor grew temperamental, when the printer ate our paper. We all learned to handle the frustrations.

Despite all the problems we encountered, I was still able, with the help of my students, to make some observations on the ways the word processor affected their writing. Three of these students, in particular, felt strongly that the word processor was valuable. Interestingly enough, all of these students spent the majority of their computer time on equipment at home. With 107 students and one computer, I just could not offer most students the time they needed to use the computer significantly in their writing. For these three students, though, the processor became a welcome replacement for paper and pencil.

Laura preferred revising on the word processor to paper and pencil because she found the processor "more efficient." Paper and pencil, she thought, presented "more of a disruption" because reading through crossouts broke her train of thought. She told me, "When I see it all out in a regular line on the computer, it sets the mood better for what's to follow. I enjoy revising with the computer. I like the neatness." For Laura, who saw herself as a "mood writer," the processor provided an easier way to keep her thoughts flowing, to keep her "in the mood" to stay with a piece of writing. I wondered if staying longer with the writing meant revising more. I asked Laura, who was already a capable writer, if she spent more time revising with the proces^sor than without it. Without hesitating, she replied, "I think the word processor makes it *easier* for me to revise, but I don't think I do any more because of it." For Laura, then, the benefit of the word processor was to make something easier that she already did.

Adam, on the other hand, had to discover a method. He'd had little experience with revision and had to be convinced of the benefits of multiple drafts—especially since he found making changes "fatiguing." The word processor, not I, did the convincing. The breakthrough for Adam came with a paper I'd assigned on *The Scarlet Letter*. His paper was a decided improvement over earlier papers he had written. He paid more attention to developing ideas and cleaned up the misspellings and punctuation errors that usually littered his writing. He credited the word processor with his success and told me that his *A* was, as he put it, "*proof* that a word processor is beneficial to the quality of writing."

He told me that because the changes he made were immediate and easily seen, he experimented more with moving text and trying new things. "On the processor you can just make a change and it's done. By hand, in order to make a change, you have to write it out and everything in between as well." Revising by hand was tiring and discouraging, and so without the processor he made few changes.

I was anxious to see if Adam's writing improvement would carry over to other assignments. On the next paper he wrote, a short story, one of the first things I noticed was how much more material Adam generated. Each time he brought in sections of his story to discuss with me, I was surprised by the increased quantity, as well as quality, of his writing. A firm believer in Donald Murray's conviction (*A Writer Teaches Writing*) that the more people write, the better writers they become, I was pleased to see Adam achieving that fluency. More important, Adam was pleased. After he completed his story, he proudly informed me that it was "the longest thing I have ever written."

When he was midway through his draft, Adam and I discussed the ways in which the processor continued to aid him in revision. In Adam's words, the processor made his writing "easier." He told me, "The *best* part, the place where it's most easy, is in the revision. It's easier to see what it is that I've just done in context with something else. I can scan through it real quickly and look for anything." For Adam, *seeing* what he wrote was important.

It was also important to Adam that his *reader* see. He talked a lot about his illegible handwriting (actually, the adjective he used was "pitiful"). "If someone else were to read it," he told me, "just as you stress it's important when you write just to write and let it flow out, anyone who's reading it has to be able to do the same—just read it rather than stutter through it. My handwriting being as bad as it is, the word processor also helps *you*, the reader, as well as me. It works both ways."

I asked Adam how the word processor was different from a typewriter. His reply came quickly: "Corrections. Editing. You don't have to retype the whole thing if you make one mistake. I know for a fact that if *I* had to rewrite an *entire* page just to make one or two corrections, I would leave them there." For Adam, I think, the benefits of the processor were interdependent, all working together—and all under the umbrella idea of *easier*.

Making corrections—editing—is an important part of the writing process, but I was more concerned about revising. I wondered if Adam were still revising the way he had with his paper on *The Scarlet Letter*. I asked him if he were making other changes in addition to corrections. "I think I'm moving *ideas* more vs. when I just did it in pencil. I pretty much kept my ideas the same way. I'd move them once in a while, but I'd get such a tangled array of arrows, as you wrote on one of my papers, 'I can't follow this.'" Again, it seemed that Adam was telling me two things about revising with the processor: one, it was *easier*, so he tended to do it more; and two, it was visually clearer, so he could better *see* what he was writing.

Blakely's writing abilities fell somewhere between Laura's and Adam's. She had some sense of process, but her writing was erratic. When she wrote papers on the processor, both she and I noticed fewer editing errors and more support. She seemed to stay with a piece longer, to "flesh it out." Like Laura and Adam, Blakely felt it was easier to see her mistakes on the screen. She often failed to notice errors on a handwritten page. I asked her, as I had Adam, why a typewriter wouldn't suffice. It could make the text easier to read, she told me, but that was all. With the processor, she could also go back and easily make changes. She found a typewriter limiting.

The greatest advantage Blakely saw to using the processor was its ability to make changes. Her writing process with paper and pen was self-defeating. "Whenever I write a paper, when I make a mistake I like to start over again right away—crumple it up in a ball and start all over." She fought a compulsion to rewrite a whole page if there were even one error on it. "With the processor I can go back and delete, insert, whatever, without having to start all over." Blakely described her revisions on the processor as "more selective" since she chose what needed to be changed and avoided reworking everything else.

These types of changes (deletions and insertions) are more in the line of "true" revision. When it came to "cosmetic" changes—editing—Blakely readily admitted that she made corrections on the processor she would not have bothered with on paper. As she explained, "When I'm reading my final copy and find something that needs revising, I can easily do it. If I didn't have a word processor, and only a few things needed changing, I wouldn't do it."

I think a comment from Adam summarizes the group's thoughts about word processing: "I can easily see how I might not have revised my report so extensively or even have put in all that I wanted to convey, were it not for the aid of that Wonderful Machine."

I am beginning to agree with Laura, Adam, and Blakely, though there were times I gladly would have given the word processor to anyone who'd take it. There were days filled with frustrations—equipment breakdowns, "crashed" programs, lost disks. But, in the end, there were the Lauras, Adams, and Blakelys who learned so much about writing, more than I alone could have taught them.

Most important, Laura, Adam, and Blakely helped us to make several observations about students who write and revise with word processors.

1. Students tend to "stay" longer with a piece of writing—adding, deleting, moving text—than they do with paper and pencil.
2. *Seeing* what they write as it appears on the screen seems to help students better determine what changes and corrections they need to make.
3. The physical act of effecting these changes is so much easier with the processor that students often take more time to revise than if they were working with paper and pencil.
4. Students working on the processor seem to find it easier to develop a sense of audience.

I think, too, that writing with the processor helped my students become more aware personally of what happens intellectually as people write. They were able to articulate clearly and decisively the methods they followed both with and without the processor. I am not sure they would have been as aware of the ways they revised had the computer not made experimenting with changes easier for them.

I set out this year to discover the ways word processing could improve my students' writing. With all of us on fairly equal footing, we learned together and our joint efforts in working through the problems confirmed for me the joys of learning *with* my students. As partners in learning, a teacher, thirty students, and a word processor taught each other a lot.

Gail G. Womble teaches at
Herndon High School, Fairfax
County, Virginia.

A Teacher's Immortality

Twelve years ago you stood petite in front of my desk
wearing your best smile,
determined dark eyes supporting your inquiry
"How can I take your English class next year?
I want to learn to write."
I used my position to schedule you in.

Wide-eyed, you birthed yourself,
I the midwife who bled your split infinitives,
your superfluous commas and contrived rhýme.
You emerged the darkened channel
dangling your modifiers—I tied them up.

Evenings you tried reforming your alcoholic father,
but you stole time to write.
Your parents didn't know you
until you presented your dad a poem—
"Wire Wizards" you called it—
Your father's CB lingo breaking over the wire
and appearing in *Mountain Review*.

I turn the pages of *The New Yorker*
and there as elegant as a rose
sits a poem "The Delicate Survivor"
by Vicky Hayes.

Shirley R. Chafin
Johnson Central High School
Paintsville, Kentucky

The Computer-Based Writing Program from Load to Print

Raymond J. Rodrigues

EVIL—But when I get the map, I shall escape from here and then the world will be very different. . . because I have understanding. . . .Of digital watches. . .and soon I shall have understanding of video cassette recorders and car telephones, and when I understand them I shall understand computers, and when I understand computers I shall be the Supreme Being.
Monty Python's *Time Bandits*

The promise of the microcomputer is that, once we understand its potential, we can develop computer-based composition curricula that will give us more choices, under our control, than we now have. Will it really make our lives easier, or is it one more educational gimmick doomed to fade into the land of Wishful Thinking along with programmed texts and opaque projectors? Early reports from composition teachers who believe in writing as a process seem promising. In fact, improvements in both technology and applications occur so rapidly these days that, by the time you read this, what I have written may well be obsolete.

Until recently, most available commercial computer-assisted programs for English were merely electronic multiple-choice tests or workbooks. Such computer software was rarely worth the cost. Within the past two years, however, composition instructors around the country have been developing interactive computer programs to lead students through the entire writing process, prewriting through editing.

A computer-based composition course using microcomputers would consist of a set of diskettes containing the subprograms for the total course. Each subprogram would guide the student through prewriting, wordprocessing, or editing. Our role would be to introduce students to the procedures employed by each subprogram, give students classroom practice in using the procedures, monitor students' use of the computer-based program, and evaluate students' work throughout the program. Because students would be working at the microcomputers for most of their writing, we would be free to work with individuals.

The key characteristic that distinguishes developing composition programs is that many are interactive. That is, the student and the computer can carry on a conversation, much as we can with a student face-to-face. The only difference is that the computer cannot think—it can only guide the student through step-by-step procedures and respond to key words it has been programmed to recognize. In some respects, we do the same; we see an unsupported generalization, so we write, "Be more specific" or "What do you mean?" We have programmed ourselves to respond to specific cues.

do the same; we see an unsupported generalization, so we write, "Be more specific" or "What do you mean?" We have programmed ourselves to respond to specific cues.

In the prewriting phase, computer programs will help students generate ideas to write about, develop supporting details for their ideas, and structure ideas prior to their beginning to write. In addition, computer programs can allow other students and us to comment on the preliminary ideas or structure at this prewriting stage, thereby providing an early audience for the student and preliminary evaluation of the actual writing.

For example, Helen Schwartz designed a program that would help students in her classes prepare a paper on a literary topic.

NAME A FICTIONAL CHARACTER X IN A LITERARY WORK.

SATAN IN PARADISE LOST

DESCRIBE THE CHARACTER X BY COMPLETING THE FOLLOWING: X IS ______

SATAN IS TRICKY

PROVIDE EVIDENCE TO SHOW THAT X IS Y: WHAT DOES SATAN DO THAT SHOWS THAT SATAN IS TRICKY?

HE ENTERS THE BODY OF A SERPENT TO DISGUISE HIMSELF. . .

WHAT DOES SATAN SAY THAT SHOWS SATAN IS TRICKY?

HE TELLS EVE LIES ABOUT THE TREE OF WISDOM TO GET HER TO EAT FROM IT.

The program continues in this fashion, asking questions to help the student develop a thesis, drawing words from the student's earlier answers to help phrase later questions. At this stage, other students can read the original responses and offer advice or raise questions before the writer continues.

Another prewriting program, one based upon creative problem-solving strategies and developed by Dawn Rodrigues and me, forces students, for example, to use analogies that they might never think of to help them generate new ideas. After the student chooses a topic and types it into the computer, the computer directs the student to pick a picture at random from a packet of pictures by the computer, for example:

WHAT WOULD YOU LIKE TO WRITE ABOUT, MARY?

ALCOHOLISM AMONG TEENAGERS

After some brief preliminary instructions about how the program works, the computer continues:

DESCRIBE WHAT YOU SEE IN THE PICTURE. TYPE A LIST OF WHAT YOU SEE, STARTING WITH NUMBER 1.

(Assume the student has selected a picture of an interior scene.)

1. *A TELEVISION SET*
2. *A RUG*
3. *A TABLE*
4. *A LARGE WINDOW*

If the student does not list anything within ten seconds, the computer prompts:

TRY TO FIND SOME OTHER THINGS IN THE PICTURE.

and the student continues listing items found in the picture.

OKAY, MARY, NOW REMEMBER THAT YOU WANT TO WRITE ABOUT THE TOPIC ALCOHOLISM AMONG TEENAGERS. I WANT YOU TO THINK ABOUT HOW THE TOPIC, ALCOHOLISM AMONG TEENAGERS, COMPARES TO A TELEVISION SET.

IF THE STUDENT USES IT TOO MUCH, IT COULD DEADEN THE MIND.

NOW THINK ABOUT HOW THE TOPIC, ALCOHOLISM AMONG TEENAGERS, COMPARES TO A RUG.

IT MIGHT COVER ALL SORTS OF PROBLEMS

After going through this process with all items, the computer asks the student to elaborate upon each and finally provides the student with a printout to take home and think about before writing.

In a pioneering prewriting program, Hugh Burns leads students through a series of structured invention questions designed to encourage open-ended responses. His program reminds students of lessons they have had previously in class and then leads them through the lesson.

I WANT TO SHOW HOW SISTER CARRIE ALLOWED DREISER TO GROW, TO CONFRONT COMMON VALUES. IT IS A MATURE WORK.

GOOD, KAREN. ADD TO YOUR RESPONSE NOW.

HURSTWOOD GROWS IN THE OTHER WAY. THERE ARE NEGATIVE PASSAGES, AREN'T THERE?

YES, THAT SEEMS OKAY TO ME.

ME, TOO.

AH, SHORT AND SWEET. PLEASE ELABORATE A LITTLE.

WELL, CHARACTERS EITHER GET WORSE OR BETTER; THEY EITHER PASS TO A BETTER UNDERSTANDING OR NOT.

TERRIFIC, KAREN! ANYTHING ELSE?

The three examples above indicate only a few of the options available, but they also demonstrate a number of characteristics of such programs. The program can act as a tutor as well as an unthinking questioner. It can encourage students with positive comments, such as THAT'S AN INTERESTING IDEA! WOULD YOU LIKE TO SAY MORE? or GOOD JOB! And as has often been noted, the computer is infinitely patient, allowing the students to experiment with ideas, to play around with words without any fear of being criticized for not being exactly on target—in short, to take risks that move writers toward greater competence.

Having developed ideas through the prewriting programs, the student can then enter into the writing, or word processing, phase. (For students who do not know how to type, schools can purchase a program that will teach typing while students play a game similar to many of the shoot-em-up, squishem-type TV games.) A number of different word processing programs are available commercially, each with strengths and weaknesses. The Bankstreet Writer, for example, is a relatively simple system to learn because students are tutored in using the program by the computer and because relatively few commands are needed to work it. On the other hand, easily learned word processing programs may be limited in their capabilities. Can they, for example, move large blocks of writing, search for and replace specific words, or find specific passages rapidly? More complex programs, such as Wordstar, which have such capabilities, take a long time for students to learn and are expensive for schools to buy.

Using a word processing program, students can do freewriting, can keep their own electronic journals, and can revise relatively easily while they write. Many students do not revise much because the act is not only time-consuming, but also physically painful, requiring much effort to recopy entire manuscripts. The word processing capability of the computer circumvents both negative aspects.

If a student bogs down in the middle of writing, that student can save what has been written and shift back to one of the prewriting programs to develop more ideas or to take an idea already developed and improve it. That done, the student can call up the original writing and continue. The relative ease with which this can be accomplished helps develop the recursive strategies of highly proficient writers.

At this point, the student might want a preliminary evaluation of the writing before continuing. In a few minutes, a hard copy can be printed out and shared among peers or shown to the teacher. Having received suggestions, the student can return to the computer to revise.

One of our most time-consuming tasks is reading student papers and writing comments. Illustrating the potential for computer-based editing progams, Writer's Workbench has been developed by Bell Laboratories for use with their hardware. We can decide which writing samples constitute appropriate models for students to follow and type those models into the computer. Such models might be excellent student writing or professionally produced, published writing—literary writing, technical writing, or whatever we desire. The Writer's Workbench then analyzes the writing according to a series of subprograms, e.g., number of *be* verbs per sentence, number of abstract nouns, and number of subordinate clauses per sentence, as well as mechanical matters such as spelling and punctuation. When a student types a draft into the computer, the program analyzes the student's writing according to the same subprograms, compares the student's draft with the data derived from the models, and, within minutes, prints out a list of suggestions for improving the writing, as the following samples show:

VARIATION

In this text 22% of the sentences are simple, 56% are complex, giving a difference of -34. This difference should range from -28 to 20 for good papers of this type.

READABILITY

The Kincaid readability formula predicts that your text can be read by someone with 9 or more years of schooling, which is a low score for [papers of this type].

DICTION

beginning line 4 (key word from title)
[in order to] come here to study my major, I had to convince both my parents and university authorities.
beginning line 6 (key word from title)
So here I sit, fat and sassy, just playing with my courses to see if I can find out whether I understand *[all of]* this.

SUGGEST

in order to: use "to" for "in order to"
all of: use "all" for "all of"

Note that the program does not revise the student's writing. The student is still free to accept the suggestions or not, to adapt those suggestions to the particular audience for whom the student is writing, and to approach us for additional help and advice. Finally, after the student has revised the draft, we can collect both the original and final drafts and the printout of suggestions, evaluating them in whatever manner is appropriate but being free to concentrate upon content more than mechanical matters.

For us to be actively involved in computer-based evaluation and editing, two main techniques are noteworthy. In the first, some computers can provide a split screen that allows us to read the student's writing on one part of the computer screen and type in suggestions and comments on the remaining portion. The student can then take the diskette with both the draft and the comments, revise accordingly, and turn the diskette or a printout of the draft back to us.

The second technique is technically simpler to devise. One time-saving capability of several microcomputers enables us to program certain keys to enter limited comments. Thus, in evaluating a student's writing on the computer screen we might hit one key and automatically type SUBJ-VERB AGR or TOPIC SENT? or whatever other comments we most commonly make. If a student needs exercises to improve specific weaknesses, we can direct that student toward electronic workbook-style exercises to work on grammar, mechanics, or whatever else is available for the microcomputer. In this context, such exercises serve a valuable purpose. In addition to using programmed keys to make specific comments, we are always free to type in additional, more detailed comments, as well as free to change those programmed keys at any time.

Admittedly a difficulty, most existing programs have been developed at individual schools, typically universities, and are not available commercially. Although it takes time, with basic programming skills, we can develop aspects of programs described here. Journals devoted to English teaching might make a valuable contribution in the near future—columns of articles containing simple programs written in Basic for teachers to copy. Even now, few programs cover the entire writing process, a notable exception being developed by Ruth Von Blum at UCLA. With the help of a grant from Exxon, she has created a program that combines prewriting, word processing, and revising. Given the limited resources of most public schools, it might be more desirable for all of us to urge commercial software companies to produce interactive programs that will enable us to teach the writing process more effectively, a promise that is easily within our reach given the current technology.

Suggested Readings

Burns, Hugh L. "A Writer's Tool: Computing as a Mode of Inventing." Paper presented at the New York College English Association Conference, October 3-4, 1980. [ED 193 693]

Burns explains how his invention program incorporates questioning strategies from three classic heuristic programs. His examples serve as excellent models for anyone wanting to develop prewriting programs.

Daiute, Colletee. "The Word Processor and Revision Strategies." *College Composition and Communication* 34 (May 1983): 149-155.

Daiute considers the physical and psychological constraints on both writers and computers and argues that working with word processors frees writers from drudgery and allows them to think.

Macdonald, Nina H., et al. "The Writer's Workbench: Computer Aids for Text Analysis." *IEEE Transactions on Communications* 30 (January 1982): 105-110.

Macdonald describes one of the most advanced editing programs available, originally developed for technical writers—a good summary of the subprograms.

Nancarrow, Paula; Donald Ross; and Lillian Bridwell. "Word Processors and the Writing Process: An Annotated Bibliography." Available from the English Department, University of Minnesota, Minneapolis 55455. ($4.00)

This is perhaps the most thorough bibliography of current information, drawing from journals, magazines, technical reports, ERIC, and others.

Schwartz, Helen. "A Computer Program for Invention and Feedback." Paper presented at the Conference on College Composition and Communication, San Francisco, March 1982 [ED 214 177].

Schwartz describes a program for literature classes in which students first write on the computer individually and then move into a "network mode" in which other students comment on what the writer has produced.

Schwartz, Mimi. "Computers and the Teaching of Writing." *Educational Technology* 22 (November 1982): 27-29.

Schwartz discusses the advantages and potential dangers of writing with a computer, considering in particular positive changes in student attitudes, such as decreased defensiveness and greater willingness to accept criticism.

Wresch, William. "Computers in English Class: Finally Beyond Grammar and Spelling Drills." *College English* 44 (September 1982): 483-490.

Wresch summarizes many of the current developments in university computer-based writing programs. See also, "Computers and Composition Instruction: An Update." *College English* in-press.

Raymond J. Rodrigues teaches at New Mexico State University.

A Writing Teacher's Shopping and Reading List for Software

Ellen K. Leahy

Finding age-appropriate software for older students has been an almost impossible task. The task is even more difficult when students have serious communication problems. Microcomputer programs have not yet been developed to remediate deficits in speaking and listening. Such software is possible, but the supportive hardware and peripherals have not been perfected and prices are out of line with operating budgets. Voice input and recognition technology is just beginning to be presented, which means that language arts software is limited to the development of writing and reading skills. Here only writing will be addressed spanning student abilities from middle school levels through college. New sources in word processing, proofreading, and text editing can be used in hands-on computer sessions, analyzing writing samples for and with students.

Drill and Practice

This category is well represented by numerous programs to improve test-taking skills for the Scholastic Aptitude Test. These programs are often accompanied by a book or manual of sample tests and the computer diskettes which offer practice on a series of similar test items. Vocabulary use, definitions, and analogies comprise the content of the verbal section of most of these programs. Other drill and practice programs cover syllabication (Pentar) and punctuation and alphabetizing (Random House).

Tutorials

In these programs, students are given a rule along with a few examples to teach understanding before proceeding with a set of examples for the application of the rule. An example of such programs would be Spelling Rules (Micro Power and Light), Lessons in Syntax (Dormac), English: Basic Mechanics (Conduit), and English Volume 1 (MECC).

Word Processing

By far the most revolutionary effect of the computer on writing instruction and the writing curriculum is the use of software for text editing. With a few key strokes, words, phrases, and entire paragraphs can be erased, recalled, moved around, and printed out. For those fortunate students with access to microcomputers, a disk drive, and a printer, the entire process of learning how to write and revise has been transformed. The key word, however, is *access* to computers. This avenue of computer use, not quite CAI (computer assisted instruction) in its purest form, is limited if the ratio of one computer to twenty-five students for a class period is the prevailing practice. A writing lab equipped with a minimum of 10, better yet 20, microcomputers, disk drives, and printers is the only realistic and cost effective way that a computerized writing curriculum can be established.

With the introduction of Bank Street Writer (Scholastic), an easy-to-learn word processing program developed and tested with elementary students under the direction of the Bank Street College of Education, the opportunity for computerizing the writing environment is now readily available. Because of its user friendly features, this program, although developed for a younger student, is now being hailed as the family word processing program. The accompanying teacher's guide suggests numerous activities which can be prepared by the teacher for student use at an "on-computer" writing session. Here is a marvelous example of a computer program which can serve multiple purposes.

A number of programs are designed to ease revising techniques. For example, Sensible Speller (Sensible Software), formerly Apple Speller, is a spelling correction program for Apple Writer (Apple Computer) and several other commercially available word processing programs. Students can obtain a frequency count of the words used in a writing sample, a total word count, and an alphabetized listing of all the words. In addition, misspelled words can be marked in the writing sample for the student to self-correct. What better way to learn to spell, particularly the demons that plague most new writers. Proofreader (Random House) checks words against an 80,000 word Random House computerized dictionary. Grammatik (Aspen Software) for the TRS-80 and CP/M based word processors checks writing style and typographical errors. It profiles a writing sample for word usage and calculates word frequency which spots overworked vocabulary. Some teachers have found a computerized readability program helpful in determining the grade level of samples from student writing. The Readability Analysis Program (Random House) displays a count of the average sentence length, the number of sentences per sample, and reading grade levels determined by several commonly accepted formulas. This type of analysis can help students see the need for expanding their vocabularies and considering sentence combining.

Prewriting Experiences and Writing Assistance

Hugh Burns' Topic Invention Program reduces writer's block by leading students through a series of questions. What disturbs some educators is that no judgment is made on students' answers. However, the value is that entering into a dialogue with the computer can solve the "I don't know what to write about" problem, which would consume a tremendous amount of time if done by the teacher for each individual in the classroom. Helen Schwartz has worked out a writing assistance program which prompts the student to generate ideas, organize thoughts, and analyze the audience for a writing project. An interesting feature is that students can obtain a printout of the prewriting process. Another Schwartz program, SEEN, assists the writer to analyze a character in a literary selection. Poetry writing can be introduced by Compupoem (Marcus), which produces poetic forms from student-supplied parts of speech. Again, the program cannot judge the correctness of a student answer. One must remember that the primary purpose of this software is communication and guidance rather than correcting or scoring answers as in traditional computer-assisted instructional programs. This may require adjusting the computer program, with the author's permission, to meet additional demands of correcting, recording, or producing hard copy, or arranging to use these programs in peer tutoring sessions.

For students who need to learn about the power of a computer and understand writing as a form of communication, the program Eliza (Artificial Intelligence Research Group) can be a worthwhile addition to an English Department's software collection. This interactive natural language game gives students a chance to experience a simple artificial intelligence program.

Another new program that should be previewed for possible use by students who have had experience with word processing is Think Tank (Living Videotext). Think Tank helps the writer organize ideas in an outline form and has the capability of letting the writer expand ideas.

Writing for the Computer

Opportunities for teachers to create writing practice exercises without programming skill are now available in a number of programs. As mentioned before, the Bank Street Writer (Scholastic) has the option of producing a series of text files. Each disk created by a teacher can hold the equivalent of 70 pages of text and can be protected from student tampering or erasing. Some examples of exercises that could be developed are writing the middle or end of a story, writing dramatic dialogue, using transitional phrases, using rhyme, writing business letters, putting sentences in proper order, a timed writing activity, changing verb tense in a passage, using synonyms and antonyms, and sentence combining.

The Teacher Utilities program, Volume 2, 3, and 4 (MECC) has the capability to produce spelling exercises and a series of drill exercises in a multiple-choice or fill-in-the-blank format. Dasher (Conduit) also gives users the option of constructing exercises of vocabulary review, scrambled sentences, sentence construction, fill-in items, and transformations. E-Z Learner (Silicon Valley) is a menu-driven program which can be used to create, store, and review questions and answers on any subject. The Game Show and Tic Tac Show (Computer Advanced Ideas) can have files set up by the teacher to drill on specific skills, rules, or vocabulary in a game format.

An excellent program, Story Board (Wida Software), permits the re-creation of a paragraph word by word from memory. It helps develop skills in memorization, short term recall, vocabulary development, and understanding of syntax. The teacher needs only to follow the simple directions for typing short paragraphs for storage on a disk. Missing Links: English Editor (Sunburst Communications) is a language puzzle that helps the student develop an appreciation of syntax, vocabulary, and mechanics of writing by allowing the teacher to create passages in which students must fill in the missing parts.

With several of these programs, students can be taught to enter their own exercises on the computer. Working in writer/editor teams, students can copy corrected worksheets to create files for future use. This type of activity, using Story Board, E-Z Learner, Tic Tac Show, Dasher, or the Game Show programs is particularly successful with students of English as a Second Language and less motivated students who need more practice in writing skills.

Programming as a New Writing Skill

Learning to write computer programs in languages of BASIC, LOGO, or PASCAL is a significant new writing skill. If one is to become a proficient program writer, a vocabulary as well as a syntax must be mastered. The integration of problem solving and writing skills is clearly needed when sequencing directions to communicate with the computer. LOGO (Apple Computer) is a microcomputer language which allows students to interact with the computer in a nonthreatening manner. It has good list-processing capabilities and allows students to be in a teaching-the-computer role.

A Final Word

Software for the English teacher and student may be advertised to serve particular instructional goals, but it should be reviewed and evaluated with the intention of documenting other educational objectives that can be attained. Software which can serve multiple purposes for teachers and students becomes a worthwhile investment. With the continuing limitations of computer programs it is crucial to explore the various possibilities of high quality software. The accompanying list is a compilation of the most promising software currently available, but how each program can meet specific student needs can only be determined by those professionals who can observe and measure their own students' learning and in relation to this prepare their students to take advantage of microcomputers as a powerful electronic writing aid.

The Shopping List

Apple Writer, Apple LOGO. Apple Computer, 10260 Bandley Dr., Cupertino, California 95014.

Eliza. Artificial Intelligence Research Group, 921 North La Jolla Avenue, Los Angeles, California 90046.

Grammatic (TRS-80 & CP/M). Aspen Software Co. P. O. Box 339-C, Tigeras, New Mexico.

Story Maker, Quill (in development). Bolt, Beranek & Newman, Inc., 10 Moulton Street, Cambridge, Massachusetts 02238.

Word Structure/College Entrance Exam Preparation. Borg-Warner Educational Systems, 600 West University Drive, Arlington Heights, Illinois 60004.

Wordy, English Grammar, English as a Second Language (in development). comPress, P. O. Box 102, Wentworth, New Hampshire 03282.

The Game Show, Tic Tac Show. Computer Advanced Ideas, 1442A Walnut Street, Suite 341, Berkeley, California 94709.

Dasher, English: Basic Mechanics. Conduit, P. O. Box 388, Iowa City, Iowa 52244.

Lessons in Syntax, Test of Syntactic Abilities. Dormac, Inc., P. O. Box 752, Beaverton, Oregon, 97075.

Writing Competency Program, English Basics—Concepts in Language Arts. Educational Activities, P. O. Box 392, Freeport, New York 11520.

PSAT and SAT Word Attack Skills. EDU-Ware Services, Inc., 22222 Sherman Way, Suite 102, Canoga Park, California 91303.

The School Tool; Grammar and Writing; Basic English Skills: Parts of Speech, Sentences; Wordwright: Dictionary diskettes for Junior High Language Arts, Senior High Language Arts, SAT Preparation, Junior/Senior High Social Studies, Elementary School. Encyclopedia Britannica, 425 North Michigan Ave., Chicago, Illinois 60611.

Writing is Thinking (IBM PC). KAPSTROM, Inc., 5952 Royal Lane, Suite 124, Dallas, Texas 75230.

SAT Preparation, Krell LOGO. Krell Software, 21 Millbrook Drive, Stony Brook, New York 11790.

Think Tank. Living Videotext, 450 San Antonio Road, Suite 56, Palo Alto, California 94306.

Crossword Magic. L & S Computerware, 1589 Fraser Drive, Sunnyvale, California 94087.

English Achievement I-V. Microcomputer Workshops, 103 Puritan Drive, Port Chester, New York 10573.

The Apostrophe, End Punctuation, Making an Outline, Compound Words, Haiku, Run-on Sentences (all PET only). Micro-ed Inc., P. O. Box 24156, Minneapolis, Minnesota 55424.

English SAT. MicroLab, 2310 Skokie Valley Road, Highland Park, Illinois 60035.

Spelling Rules. Micro Power & Light, 12820 Hillcrest Road, Suite 224, Dallas, Texas 75230.

Language Arts Packages. Milliken Publishing Co., 1100 Research Blvd., St. Louis, Missouri 63132.

English Vol. 1 (Parts of Speech); Teacher Utilities (Vol. 2, 3, 4); Teacher Utilities (Vol. 1). Minnesota Educational Computing Consortium (MECC), 2520 Broadway Drive, St. Paul, Minnesota 55113.

Compupoem. Stephen Marcus, South Coast Writing Project, University of Southern California, Santa Barbara, California 93106.

Writing Invention Program (developed by Hugh Burns, and adapted for Apple as TOPI program). Denise McGinty, RASSL/Learning Services, University of Texas at Austin, A332 Jester Center, Austin, Texas 78784.

Match Game. Opportunities for Learning, Dept. L-4, 8950 Lurline Ave., Chatsworth, California 91311.

Ask Me Why—Syllabication. Pentar Software, 87 Fern lane, Newton, Connecticut 06470.

Analogies, Story Builder, Vocabulary Builder, New Step by Step, Step by Step Two. Program Design, Inc., 11 Idar Court, Greenwich, Connecticut 06830.

Story Builder, Story Starter (TRS-80), Proofreader (IBM PC), Fundamental Punctuation Practice, Sequence and Alphabetizing, Readability Analysis Program, Random House School Division, Dept. 985, Suite 201, 2970 Brandywine Road, Atlanta, Georgia 30341.

Glossary of Usage, Word Catagories. Right On Programs, P. O. Box 977, Huntington, New York 11743.

Apple Speller, Sensible Speller. Sensible Software, 6619 Perham Drive, Dept. M. West Bloomfield, Michigan 48033.

Story Builder, Vocabulary 12th Grade, Poetry Writing, Bank Street Writer. Scholastic, 730 Broadway, New York, New York 10003.

Wordskill for the Microcomputer, Cross Clues. Science Research Associates (SRA), 155 N. Wacker Drive, Chicago, Illinois 60606.

E-Z Learner. Silicon Valley Systems, 1625 El Camino Real #4, Belmont, California 94002.

Missing Links: Literary MicroAnthology, English Editor. Sunburst Communications, 39 Washington Ave., Room T1616, Pleasantville, New York 10570.

SEEN (literary analysis program), Writing Assistance Program (in development). Helen J. Schwartz, Dept. of English, Oakland University, Rochester, Michigan 48063.

Story Board, Questionmaster. Wida Software, 2 Nicholas Gardens, London W5 5HY.

The Reading List

Bradley, V. N. "Improving Students' Writing with the Microcomputer." *Language Arts* 59 (October 1981).

Burns, Hugh. "Pandora's Chip: Concerns About Quality CAI." *Pipeline* (Fall 1981).

Burns, Hugh L. and George H. Culp. "Stimulating Invention in English Composition Through Computer-Assisted Instruction." *Educational Technology* 20 (August 1980).

Dauite, Collette. *Computers and Writing*. New York: Addison Wesley, 1983.

Franke, Thomas L. "Computers and Writing Instruction: Issues for Policy Makers." *Pipeline* (Spring 1982).

Grady, M. Tim and Jane D. Grawronski (eds.). *Computers in Curriculum and Instruction*. Reston, Virginia: ASCD 1983.

Hennings, D. G. "Input: Enter the Word-Processing Computer." *Language Arts* 58 (January 1981).

Kepner, Henry S. Jr. "Computers in the Classroom." Washington, D.C.: NEA, 1982.

Lawler, Joseph (ed.). *Computers in Composition Instruction*. Los Alamitos, California: SWRL Educational Research and Development, 1982.

Marcus, Stephen. "Compupoem: A Computer Assisted Writing Activity." *English Journal* (February 1982).

Marcus, Stephen. "The Muse and the Machine A Computer and Poetry Project." *Classroom Computer News* (Nov./Dec. 1982).

Rubin, A. "Making Stories, Making Sense." *Language Arts* (March 1980).

Schwartz, Helen. "But What Do I Write? Literary Analysis Made Easier." *The Computing Teacher* (August 1983).

Southwell, Michael, Mary Epes, and Carolyn Kirkpatrick. "Computer-Assisted Sentence Combining." Paper presented at NECC 1983 Conference, Baltimore, Maryland, *Proceedings of 5th Annual National Educational Computing Conference*. Iowa City: University of Iowa 1983.

CRLA (*Computers Reading and Language Arts*, a professional journal). P. O. Box 12039, Oakland, California 94661.

Ellen K. Leahy teaches at Bronx Community College, New York.

APPENDIX

3

USING DATA BASES IN SOCIAL STUDIES

One important outcome of a balanced social studies program is for students to learn to gather, organize, analyze and use information in order to become better decision makers, problem solvers and planners. Goals related to these capacities are repeated at each level of the social studies curriculum, K-12. Selected skill goals for Social Studies are listed below.

GOAL 1:	*The learner will identify and define problems and suggest ways to solve them.*
GOAL 2:	*The learner will locate and gather information*
GOAL 3:	*The learner will evaluate information*
GCAL 4:	*The learner will organize and analyze information, and draw conclusions.*

Each of these goals can be developed and reinforced as students become involved in activities that either require them to make a decision, solve a problem or plan. A data base program provides the means to make comparisions that will show trends reflected by changes in data over time.

Exercises of the type found in these materials can provide needed practice in organizing and analyzing data. The computer and data management software allows students to make comparisons and see relationships they would not otherwise see.

The sample problems in these materials can be solved using information that is easy to find. The teacher's imagination and initiative are the only limits to the type of research projects, decisions, and problems that students can tackle using a data base program.

The use of a data base management program allows a class to share the responsibility of locating and entering large amounts of information for use by the entire class. Without the data base program such information would be unavailable in such a useful form. It is the teacher's responsibility to determine which units of study can be enhanced by using a data base.

Information by itself is of little value. However, appropriate information can be invaluable when faced with a problem to solve, or an important decision to make. With the benefit of a computer and an appropriate data base program, massive amounts of information which would baffle most students, becomes more coherent, manageable, and understandable. In order to become better decision makers, students need the benefit of working with large amounts of information in an understandable fashion. They live in a time when they will be confronted with increasing amounts of often complex information. The use of computer data base programs in the social studies classroom can contribute to the development of important skills that will help students face this challenge. The approach introduced by these materials will also teach the value and limitations of the computer.

COMPUTER COMPETENCIES

In the Library/Media and Computer Skills section of the NORTH CAROLINA STANDARD COURSE OF STUDY and the TEACHER HANDBOOK, the following goal for computer awareness is given at all grade levels:

GOAL 6: The learner will demonstrate an understanding of computers, their operation, and their possible application to solving relevant problems.

In grade seven, one of the objectives for this goal is:

6.2 Understand the uses of a computer to process data.

Proficiency in this objective could be obtained through the use of non-computerized activities, field trips, interviews, and media. Students do not necessarily need to have hands-on experiences with a computer.

However, by grade eight the objective has changed slightly to read:

6.2 Demonstrate an increasing proficiency in the use of a data base program.

It is suggested that at this grade level students should use a computerized data base program in order to understand fully its operation and advantages.

Even though these two objectives are introduced at grade seven and eight, the skills obtained should be practiced and used thereafter. This can best be accomplished through an application to the instructional program.

INTRODUCTION TO DATA BASES

WHAT IS A DATA BASE?

DATA BASE: **A collection of information on a particular subject organized in a specific manner.**

Collections of information--words, numbers, symbols--exist in many different forms. In most homes there are collections of recipe cards and collections of addresses; in school there are file cabinets of student work and collections of catalog cards in the media center card catalog. The telephone book and the dictionary are also examples of collections of information. In the world of work, collections may consist of car license tag numbers, or airline schedules or store inventory lists. These collections are used to find specific information and to help make decisions.

Microcomputers allow greater flexibility in using collections of information. The information can be rearranged and changed to find specific information more easily. This flexibility provides more accurate information and provides it faster enabling people to make better decisions. The use of the microcomputer to manipulate these collections has resulted in a new term for collections of information. The collection is called a data base and the information in the collection is called the data. The microcomputer program which contains and manages the data is called a data base management program or a file handler.

How does a data base management program relate to the instructional program of a school? Unlike many computer assisted instructional programs, a data base management program is not a tutorial, a simulation or drill on facts. It does not "present" information for the student to learn. Instead, it is a utility to facilitate the activities of organizing, manipulating, and accessing information relevent to the current curriculum. Skills of sorting information, of searching for information by keywords, of outlining facts, and of analyzing information to make decisions are not new skills. Students are taught these essential skills starting in kindergarten. However, these competencies can be strengthened by the use of a data base management program.

In this age of technology, the ability to select, organize, and manage information are important skills. Although important, these skills are not ends within themselves. The benefit of using data bases in the instructional program is the development of decision making skills.

"Limited-use" data bases

Data bases are all around us. Some data bases are a collection of information which is already arranged in a certain manner. An example of this type of data base is the telephone book. Think about how a telephone book is organized: the information is grouped by cities; the information in each grouping is arranged alphabetically by the last name of an individual or by a company name. We can **SEARCH** the telephone book for the phone number or the street address of a specific individual or company. We can obtain a listing of all individuals in the phone book with the same last name. However, it is not as easy to locate an individual with the phone number of 365-4092 or to select all individuals living on Edgewater Drive. Ways to use this type of data base are limited. The type of data in a telephone book is a "limited-use" data base. Other examples of "limited-use data bases are: recipe files, address books, checking account statements, grocery lists, newspaper want ads, television schedules, department or discount store catalogs, warehouse inventories, restaurant menus, sports team statistics, record store album lists, media center card catalogs, class grade books. One characteristic of a "limited-use" data base is that the the information is printed on paper. This creates a problem if the data base needs to be sorted. We must rearrange or even cut the pieces of paper in order to **SORT** the data base differently. **SEARCH**ing these data bases for specific information is not as much of a problem. However, it can be time-consuming and prone to errors.

SEARCH and SORT--these are two main operations on a data base. These terms are defined below.

SEARCH: **An operation to select data from the data base according to specified criteria or conditions.**

SORT: **An operation to rearrange the order of the records in the data base according to a specified alphabetical or numerical order.**

"Multi-use data bases

Data bases range in size from very small lists to enormous amounts of information such as the U.S. census. Think of some other very large data bases. (Examples: voter registration records, hospital patient records, automobile license tag registrations, armed forces files, bank records) It takes a long time to SORT and SEARCH paper versions of these large data bases. Not only is there time involved in a problem but also the accuracy of the work. Thus, the relationships between many of the items of data in these large data bases is seldom discovered.

The increased availability of the microcomputer is allowing people a chance to work with large data bases. By entering the data into the computer, they can take advantage of the speed and accuracy of the computer to manipulate the information to determine possible relationships. Therefore, they can collect more data for better decisions. These electronic data bases are called "multi-use" data bases. They can be sorted and searched in many more ways and much faster than "limited-use" data bases. The two main operations of a "multi-use" data base are stili SORT and SEARCH, but the parts of these data bases have specific names. Carefully read the following definitions and examine the diagram below to learn some of the parts and functions of a multi-use data base:

FILE: **The name of a collection of information stored on a microcomputer diskette or loaded into the microcomputer.**

RECORD: **A unit of data in a data base consisting of all the data in the specified categories for that unit.**

FIELD: **A category in which data is included for a particular record.**

CHARACTER: **A number, letter, or symbol used to form the data.**

ADD A RECORD: **A function used to enter information into a data base.**

EDIT/UPDATE A RECORD: **A function used to change the data in a record.**

DELETE A RECORD: **A function used to eliminate a record from a data base.**

PRINT: **A function used to produce a paper copy of the selected data.**

BROWSE: **A function used to scan the data in a record or the records in a file.**

ILLUSTRATIONS BY ROBERT CONRAD

Get Organized for Good With Data-base Software

CREATE A COMPUTERIZED FILE CABINET THAT PLACES INFORMATION AT YOUR FINGERTIPS

BY STEVE MORGENSTERN

Don't look now. but your home is crawling with . . . data!

Yes. it's true: even in the most nontechnical home. there's data in every nook and cranny. Over there—in the desk drawer. That envelope crammed with receipts. What a load of data! And under it. your insurance policies.or your tattered address book. or the membership list from the P.T.A. . . . it's a whole drawer simply filled with data.

Surely we can find some place without data. But no—in the kitchen there's a calendar. with data about birthdays and anniversaries scrawled here and there. How about the kids' rooms? Even here. data is rampant. Look at that checklist of baseball cards in Junior's collection. or the notes for Annie's school report about snakes.

It's time to face facts—"data" is nothing but information. and the amount of information we all deal with every day just keeps growing. What can we do about this dizzying deluge of data? We can tame it. organize it. and capture it in a form in which we can control it. That's where data-base software comes in.

YOUR DIGITAL FILING SYSTEM

You'll find literally dozens of programs designed to help you file information of all sorts. Some are called "filing programs." others. "data bases" or "data-base managers." For our purposes. the terms mean the same thing (see *"Relational? What's That?"*).

You could spend $600 or more for a database program designed for business use. Filing managers for home or small-business use.

on the other hand, are available for anywhere from $20 to $200.

Within this price range, there are plenty of titles from which to choose for most computers. Lots of choices means lots of decisions, so you'll need to know the right questions to ask when sorting through the alternatives. With that in mind, let's take a question-and-answer approach to the field of filing programs, starting with a very common question.

WHY BOTHER?

As we tried to indicate, most people have some kind of information they need to keep handy. Granted, there's nothing you can put into your filing program that you couldn't write down on paper. *The real difference comes when you try to find something.* Filing software gives you a way to make sense of your data—to put it in order, and pull the one tidbit you need from a mountain of information.

Take, for example, my taxes. For most of last year, my high-tech solution to this national problem was a shoebox. Then, as April 15 rolled closer, I took shoebox in hand, booted up a friendly data base, and entered the information from a year's worth of receipts into a computer file. I did not type the information in any particular order, or presort the scraps; I just sat down and typed away. Then, when all the information was entered, I could ask for all my postage receipts in chronological order, for instance, and print out the results. The same went for travel expenses, income, and interest earnings. All of it went jumbled into the computer; it came out neat, ready to be filled in on the IRS schedules and forms.

A data base listing your household possessions and their value can be extremely helpful if you ever have to file an insurance claim. The same thing is true with all your hardware and software purchases. A family medical history is another easy and potentially valuable use of a filing program (*see the Home/Money Management column in the November 1985* FAMILY COMPUTING).

Seen as general purpose tools, filing software can answer many needs that might otherwise be filled by specialized packages. This can include home budgeting or a checkbook register, tracking stock and bond portfolios or recipe files, and organizing research notes for term papers or even doctoral dissertations.

WHAT'S A FILE, AND HOW DO I SET ONE UP?

While there's certainly plenty of variety among filing programs, the basic procedure for putting information in and getting it out again is fairly standard.

In setting up, decide which categories you want to include (like PHONE in an address file). Name each category, entering that name into the filing program to create a form—just like a blank form you might type out if you were creating a file on paper. Then, each time you enter the data, you'll simply fill in the blanks.

The jargon is simple—with a data base, categories of information are called "fields," and a blank form is a "record." Together, all your records on one subject constitute a "file." Each record might contain data about one person in an address file, for example, or one source in your reference notes. Within every record there are separate fields for each category. In an address-file record, for instance, you'd likely have a NAME field, a PHONE field, a ZIP CODE field, and so on.

THE DATA DANCE—ARRANGING, CHANGING, AND REARRANGING

Imagine things the old way. Each record in your address file is on an index card. You can reshuffle the deck of cards into any order, according to any category on the card. Alphabetical order by last name? Birthday order? Take your choice—but be prepared to spend some time and effort reading and rearranging each card.

Now boot up a filing program. All of a sudden, it's easy to order your facts, to rearrange them whenever it suits you. This is the "sort" function, an especially handy tool when you're adding new information or changing old. Instead of trying to find the right spot for this new data, you simply add it to the file. Invoking your program's sort command makes all the data fall into place in your chosen—perhaps alphabetical—order.

Every filing program we've seen has some kind of sorting capability. At the simplest level, you can put words into alphabetical order or numbers into numerical order. For instance, a program such as *Friendly Filer* will sort information only from A to Z or smallest number to largest. If you want your file arranged so the bigger expenses come before the smallest, you're out of luck.

On the other hand, software like *Bank Street Filer* lets you sort from lowest to highest or from highest to lowest in at least three ways: alphabetically, numerically, and chronologically.

HOW DO I FIND SPECIFIC INFORMATION?

Finding a particular bit of information within your file is accomplished by a program's "search" feature. One example of a search feature is exact matching. If I'm looking for all the people named "Smith" in my file, I will ask the program to find every record with the name "Smith" in the LAST NAME field.

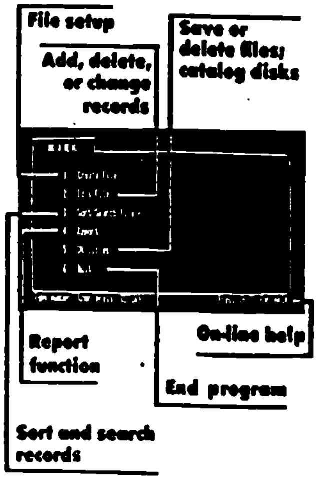
There are some kinks to consider. If you set up your data base with separate fields for LAST NAME and FIRST NAME, then you can pull out the Smiths with an exact match search under LAST NAME. But if you've used a NAME field and entered "John Smith," "Jane Smith," and so on, then "Smith" is not an exact match for either one. You'll need a different method for searching.

A MENU-DRIVEN DATA BASE

In data-base programs (as in many other types of software), you'll control the program in two main ways: through menus or through commands. With menu-driven software, you activate the program's functions by making choices from a menu. With the command-driven type, you need to remember what key(s) to press for any particular function. Menu-driven is easier (nothing to memorize); command-driven is faster (no menus need to be printed on-screen). Some filing programs (such as the data-base module in AppleWorks*) combine menus for certain functions, with commands in other sections.*

The representative screens on this page and the next few pages illustrate the menu-driven concept.

MAIN MENU



STEVE MORGENSTERN *is a software reviewer for* FAMILY COMPUTING. *This is his first feature article for the magazine.*

SELECTED FILING SOFTWARE FO

On these two pages, you'll find a selection of filing software packages suitable for use by most families. In a few cases, data bases listed are powerful enough for small businesses; that's noted where appropriate. An explanation of the terms and criteria used in the chart follows.

Hardware Requirements: Unless otherwise stated, minimum memory requirements are 48K for Apple II series, 48K for Atari, 128K for IBM PC/PCjr and compatibles, and 128K for Macintosh. Of course, versions of the same title for different computers may vary.

Max. No. Records, Max. No. Fields, and Max. Size Record: The three "max" categories are interrelated. As the values in either of the last two go up, the maximum number of records in one file goes down. To wit: Even though the *AppleWorks'* data base allows each record to contain up to 1,024 characters, one file cannot hold the full 1,350 records if each record consists of the maximum number of characters. As you set up more fields, your **Max. No. Records** decreases, too. (**Max. Size Record** is measured here in characters.)

Sort Types: "Alpha" means to sort mixed text (words alone in a field, or words and numbers); "numeric" refers to number sorts only. "Ascending" is A to Z or zero to nine, while "descending" is the reverse: Z to A or high number to low number.

Search Types: In order to best use limited space, the terms employed in this column are in verbal or symbolic shorthand. For instance, "match" refers to two different, but related methods of finding information: exact and embedded matches. Some match searches are also case-sensitive (upper- or lowercase). The numeric equivalent of match is "=". "Not match" is the same as "not =", but applied to words, rather than numbers. When applied to words, the numeric searches ">" (greater than) and "<" (less than) become a "range" search. A "wild card" search accepts any character in a given posi-

Title/ Publisher/Price	**Hardware Requirements**	**Max. No. Records**	**Max. No. Fields**	**Max. Size Record**	**Sort Types**	**Search Types**	**Entry/ Speed**
AppleWorks Apple Computer (408) 996-1010 $250	Reviewed on Apple IIc. Also for Apple IIe.	1,350	30	1,024 char.	Alpha/numeric/ date: ascending/ descending.	Match and not match. Range. Numeric: =, not =, >, <.	Entry extremely easy; speed unusually fast.
Bank Street Filer Broderbund Software (415) 479-1170 $50–$70	Reviewed on C 64/ 128. Also for 64K/ 128K Apple.	255	50	4,000 char.	Alpha/numeric/ date/time: ascending/ descending.	Match and not match. Numeric: =, not =, >, <.	Complete menus and prompts make it simple and quick.
Better Working: File & Report Spinnaker (617) 494-1200 $50–$60	Reviewed on Apple. Also for C 64/128.	2,000	20†	1,600 char.	Alpha/numeric: ascending/ descending.	Match and not match. Wild card. Numeric: =, not =, >, <.	Data entry relatively easy and reasonably rapid.
The Consultant Batteries Included (416) 881-9941 $59–$100	Reviewed on C 64. Also for C 128, IBM PC/PCjr.	Limited by disk capacity.	99	98,010 char.	Alpha/numeric: ascending.	Match. Wild card. Numeric: =, not =, >, <.	Good method of data entry. So-so speed.
Data Manager 2 Timeworks (312) 948-9200 $50–$130	Reviewed on C 64. Also for C 128, IBM PC/PCjr and compatibles.	2,000	20	230 char.	Alpha/numeric/ date: ascending/ descending.	Match. Record no. Numeric/date: >, <.	Setup OK, but slow in use.
Data Perfect LJK Enterprises (314) 962-1855 $130	Reviewed on Apple. Also for Atari.	3,000	32	4,224 char.	Alpha/numeric/ date: ascending/ descending.	Match and not match. Wild card. Numeric: =, not =, >, <.	Fast setup with handy features. Quick, with data in RAM.
E-Z Base Spectrum Projects (212) 441-2807 $25	32K Tandy Color Computer	500	15	256 char.	Alpha/numeric: ascending/ descending.	Match.	Setup and entry easy; slow in operation.
"I Know It's Here Somewhere!" Hayden Software (617) 937-0200 $60	Macintosh.	Limited by Random Access Memory (RAM).	20	16,000 char.	Alpha/numeric: ascending.	Exact match. Wild card.	Easy to set up and use. Relatively quick.
The Manager Commodore (215) 431-9100 $50	C 64/128.	2,000	250	1,500 char.	Alpha/numeric: ascending/ descending.	Match. Numeric: =, not =, >, <.	Good edit functions for record creation.
MasterType's Filer Scarborough (914) 332-4545 $40	Reviewed on Apple. Also for Atari, C 64/128, IBM PC/PCjr.	250	11	330 char.	Alpha/numeric: ascending/ descending.	Match and not match. Numeric: =, not =, >, <.	Easy to do; moderately quick.
Microsoft File Microsoft Corp. (206) 828-8089 $195	Macintosh.	65,535	1,023	Limited by disk capacity.	Alpha/numeric/ date: ascending/ descending.	Match and not match. Range. Numeric: =, not =, >, <.	Preset record format makes entry speedy.
PC File Computer Easy (612) 829-9614 $20	IBM PC/PCjr and compatibles.	2,000	42	1,365 char.	Alpha/numeric: ascending/ descending.	Match and not match. Wild card. Numeric: =, not =, >, <.	Easy entry, but limited format. Operates quickly.
PFS:File Software Publishing (415) 962-8910 $50–$175	Reviewed on 128K Apple. Also for C 64, IBM PC/PCjr, Macintosh.	32,000	32,000	Limited by disk capacity.	Alpha: ascending.	Match and not match. Range. Wild card. Numeric: =, not =, >, <.	Data entry is quick. Acceptable speed in use.
SynFile+ Synapse/ Broderbund (415) 479-1170 $50	Atari.	13,000	66	255 char.	Alpha/numeric: ascending/ descending.	Match and not match. Range. Wild card. Numeric: =, >, <.	On-screen prompts make setup/entry easy. Fast-running.

KEY—Numeric search types: = Equal; **Not** = Not Equal; > Greater than; < less than. **ON** On-line help screens.

FAMILY AND SMALL-BUSINESS USE

tion. For example, in a search for state abbreviations (where a question mark is the wild card), "N?" will deliver NH, NY, NJ, NM, ND, NC, NV and NE. (*See the main text for a complete explanation of searching.*)

Entry/Speed: The information in this column has been subjectively determined by our reviewer. "Entry" refers to the ease of use (or lack of ease) in entering information. "Speed" means "How fast does this program operate (especially in the three areas of sorting, searching and data entry)?"

Change Files/Records/Fields: What happens when you want to add or delete records after a file is created? Can you change records without disturbing the existing data? Is it possible to copy, move, or combine files, records, or fields? Look in this column for answers.

Report and Math Functions: Three major types of report formats are listed. "Tabular" means a format with lines (or rows) and columns. In almost all cases, math functions demand a tabular format; most calculations are performed on a column, rather than on a line. The "list" report format can refer to a simple, top-to-bottom listing and/or a more complex, "free-form" page layout. "Labels" refers to mailing labels, usually one per line. "Printer codes" means you can send special control characters to your printer in the report mode. This allows a better command of your printer's particular features, including the compressed typeface.

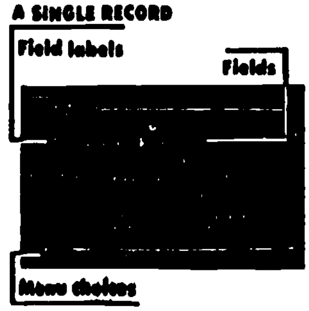
"Four-function arithmetic" is a fancy way of saying addition, subtraction, multiplication, and division. "Logical functions" refer to "if/then" kinds of statements (*see the bowling example in main text*): this power is often found only in business-class data-base software.

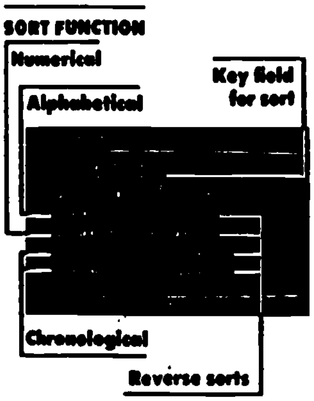
Documentation: Like **Entry/Speed**, this is a reviewer's opinion category, as are the four columns of **Ratings**.

Further Notes and Comments: Here you'll find more facts (such as file compatibility), and additional thoughts from the reviewer.

Change Files/ Records/Fields	Report and Math Functions	ON	Documentation	RATINGS O	 EU	 EH	 V	Further Notes and Comments
Can add/delete fields after file is made, but you lose record formats. Makes subfiles	Types: tabular, lists, and mailing labels. Totals, subtotals, 4-function arithmetic. Printer codes.	Y	Takes you from disk tutorial, to manual, to reference materials—a worthwhile arrangement.	★★★★★	★★★★★	★★★★	@	Integrated software, including word processor and spreadsheet. Shares files with *VisiCalc* and most word processors. Good for business and home use.
Can change field names, types, and positions after file is made. Combines existing files.	Types: tabular, lists, and mailing labels. Totals, subtotals, 4-function arithmetic.	Y	Excellent disk tutorial lets you start without looking at good manual.	★★★★★	★★★★★	★★★★★	★★★★★	Enough power for most users, and very friendly throughout. Can append up to 7 lines of comments to each record. Shares files with *Bank Street Writer* and *Mailer*.
Cannot add/delete fields after file is made. Can copy record layout to new file.	Types: tabular and labels. Totals, subtotals, averages, max./min., 4-function arithmetic.	Y	Manual looks better than it reads; has weak instructions, such as its explanation of sorting.	★★	★★★	★★★	★★★	Flexible search function, but can't save report layout and criteria—a big flaw. Shares files with *Better Working* word processor and spreadsheet.
Add/delete fields after file setup. Can create new file from existing data.	Types: lists and labels. 4-function arithmetic, record count. Printer codes.	Y	Some poor explanations detract from manual. Could explain in more depth.	★★	★★	★★	★★	Shares files with *PaperClip* word processor. Password protection available. Cannot back up single files—full disk only.
Add/delete fields to existing records. Can also change record sequence.	Types: lists and labels. 4-function arithmetic. With graphing feature.	N	Must jump back and forth between sections, and no index. Weak graphing explanation.	★★	★★	★★	★★	Shares files with *Word Writer*, *SwiftCalc*. Best for business: password mandatory. The graph function only special feature.
Add/delete/insert fields after setup. Can transfer selected info into new data base.	Types: tabular and labels. 4-function arithmetic, plus LOG, INT, ABS, EXP, and SQR math functions.	N	Clear reading in an extensive, step-by-step tutorial—a real strength.	★★★★	★★★★	★★★★	★★★★	Shares files with *Letter Perfect* word processor. For business as well as home. Helpful built-in calculator for entering numbers. Nice for novices and experts.
Add/delete fields, or change field names.	Types: none. Printer codes. No math functions.	N	Covers all the necessary information, but nothing more. Offset quality.	★★	★★★★	★	★★★	A good filing program for kids or beginners, but is limited in use (i.e., no report function). Shares files with *Telewriter-64*.
Cannot add/delete fields after data entry. Can change field size or field label.	Type: lists only. No rea' math or logical functio	Y	A manual that is clear enough, but you wish there were more information.	★★	★★★	★★★	★★★	Record size limited to one screen. Good enclosed sample files. Can have 5 files open at one time in separate windows. A beginner's file program.
Add/delete or rearrange fields after data entry. Can change files.	Type: lists only. Totals, 4-function arithmetic. If/then, and/or logic.	N	Skimpy and abstruse manual gives short shrift to program's powerful features.	★★★	★★	★★★★	★★★	For sophisticated home users or small businesses. Arithmetic calculations can be used for "what-if" planning.
Cannot add/delete fields after file is made. Can make new file based on data in original file.	Types: tabular, in lines as well as columns. Totals, but no logical functions or printer options.	N	A strong point of the package. The manual is clear and helpful, with cartoon illustrations.	★★★★	★★★★★	★★★★	★★★★	40-column display; 80-column printout. Good for children or adults who want an easy program. A limit of 10 files per disk.
Can add/delete fields after file is made. Easy to add data to existing records.	Types: tabular, lists, and mailing labels. Totals, averages, max./min., standard deviation, count.	Y	The manual is comprehensive, logically arranged, and has a good index.	★★★★★	★★★★★	★★★★	★★★	Can include *MacPaint* images in file. Shares files with *Word*, *Multiplan*, *Chart*, and *MacWrite*. Reference card enclosed. Works well in offices or homes.
Add/delete fields. Change field names, but not positions. Can copy, rename, merge files.	Types: tabular and labels. Totals, subtotals, and 4-function arithmetic.	N	The manual gets the job done, but nothing more.	★★★★	★★★	★★★	★★★★★	Straightforward filer, with some surprisingly powerful features. Includes password security. For business as well as home use.
You can do major redesign after file is made. Create subfiles and merge file data.	*PFS:Report* bought separately. Types: tabular and labels. Totals, subtotals, averages, 4-function arithmetic.	Y	Well-structured, with good examples. Aimed at mature user.	★★★★	★★★	★★★★	★★★	Shares files with other *PFS* titles. Especially useful with text-heavy files, thanks to flexible sizing of fields and records. For home and office. (Mac comes w/*PFS:Report*.)
Add/delete fields. Create subfiles and merge file contents.	Types: tabular and labels. Print codes. Totals. 4-function arithmetic, plus more math functions.	N	Complete tutorial on disk and a well-presented manual make this non-intimidating.	★★★★★	★★★★	★★★★	★★★★★	Shares files with *Atari Writer*, *SynCalc*, *VisiCalc*, *PaperClip*. A top-notch data base, with sophisticated features and good prompts for user.

RATINGS—**O** Overall performance; **EU** Ease of use; **EH** Error-handling; **V** Value for money; * Poor; ** Average; *** Good; **** Very good; ***** Excellent; @ Excellent for data base only (if you need the integrated word processor and spreadsheet).





CONSIDER WHAT'S AVAILABLE: CHOOSING YOUR DATA BASE

The accompanying chart gives the "vital statistics" on more than a dozen filing programs. but that's just the tip of the iceberg. We chose these programs to provide a representative sample of what's out there. You'll find many more choices on the store shelves. Therefore. to help you make an informed selection. here are some guidelines for choosing a data base:

WHO WILL USE IT?

You'll find filing programs simple enough for everyone from about 8 years old and up. while others are highly technical. One key difference is the program's structure. Will you have to remember commands when you want to do something. or can you make choices from an on-screen menu? Working with a menu is certainly simpler. but it's often accompanied by less power.

EASE OF USE VS. POWER

Some of the simpler filing programs are so easy. you can master them in the course of a day. Others take weeks to understand all their capabilities—but if you need those features. that's time well spent. Of course. price is linked to power as well, although we found some very muscular data-base packages at very reasonable prices.

WHAT WILL YOU USE YOUR DATA BASE FOR?

What purpose will you 'ata-base files serve? Odds are. you will have cific task in mind when you decide to buy a data base. Will any program you're considering accommodate the type of data you'll be entering? Can it handle a large enough file? Will its search. sort. and report capabilities let you conveniently pull out information in the form you need it?

Consider all the ways you might possibly use the program in the future. Paying a few extra dollars for more flexibility now may be rewarded by greater use later on.

READ THE MANUAL

Take a look at the documentation. The fastest way to get a good picture of a filing program's capabilities is seeing it in black and white; it's better than a knowledgeable friend's well-meaning explanation. or a salesperson's quick pitch. Of course. if you can borrow a program from a friend and work with it for a while before purchasing. that's even better.

A significant question involves uppercase and lowercase letters. Usually. you are free to enter data in any mixture of capital and small letters you choose. But that can affect the way you search for the information later. since some search functions are case-sensitive. In a case-sensitive search for "Smith." the listings for "SMITH" or "smith" won't appear.

Another type of search looks for an embedded sequence of letters anywhere they occur. If you've put first and last names into a single NAME field. you could find all the Smiths by searching for the name wherever it exists. in that field or in the whole file.

Still another kind of search uses wild cards. A wild-card search accepts any letter or number in a particular position. For example. if an asterisk serves as a "wild card" in your program. then searching for "06/**/86" will give you all the files dated June 1986.

Another common way of finding dat range search. using "greater than" or than" (such as. find all the names greater M. meaning those that start with letters M to Z). There also are "not equal to" sca (find all states that aren't Iowa. for exam

What makes searching especially pow is the ability to check more than one fiel record at the same time. This is a form of or" logic. If you wanted to find all the peo your file whose last names start with W. live in Nebraska. and who are more tha years old. the better data-base packages le pick out only those records that satis three conditions.

HOW DOES A DATA BASE DELIVER THE GOODS?

Any filing program we've tested let choose between reading the information request on your monitor and having it pr out on your printer. Differences arise in options as viewing one record at a tim screen or several at once.

In addition to recreating the blank f you've filled in. many programs let you the information in the form of labels. or simple line-by-line listing. When you ge fancier formats. however (for instance. outs which include only some of the fiel each record). then you are looking at a tion called "reporting."

Report functions begin with designing printout or the on-screen data display. example: Does the phone number come the name or after the address in an ad file? It's usually your choice. Within the li tions imposed by your program and your pr you can specify where you want each it information printed. and what kind of ings you want for a column or report sec

Math capabilities are often part of t port feature. They're very useful. When I u data base to help with my taxes. I ha program print a total of all my expense in each category. Some report function vide only a single total at the end of a r others let you construct complex mathem formulas.

HOW MUCH MATH WILL A DATA BASE DO?

One general difference between the tively simple filing programs we've exam and expensive. business-oriented data managers is the complexity of the mathe cal functions available. Even at the lowe of the price scale. though. you can find sophisticated capabilities.

The Manager from Commodore is package that lets you create fields that co data calculated from information found i er fields. Let's say you put the scores o bowling team into a file with *The Man* You could include a field in each team ber's record called "Average," and enter a formula for calculating averages. The program vould automatically update this figure when you add each week's scores.

That just scratches the surface of mathematical reporting capabilities, though. *The Mana;er* actually lets you write short programs vithin the data base, basing its calculations and printouts on logical and arithmetic calculations. For instance, if a bowler scores over 200, then place his or her name in the honor-roll field.

These kinds of mathematical and logical abilities provide a lot of power, but they're not always easy to use. If you think you'll want to perform complicated manipulations of your information, then pick a filing program that offers these functions.

DO I HAVE TO PLAN AHEAD?

Most simply, the amount of planning re-quired before creating a data-base file depends on the software you've chosen.

Generally speaking, you can add new fields to a file (up to its maximum capacity) even after you've started entering information, but you won't be able to move the fields around. That is, if you've created a file with NAME, ADDRESS, and PHONE fields, and you later decide vou'd like an AGE field, you'll be able to add AGE it the end of each record, but you won't be able o tuck it in between ADDRESS and PHONE.

You may have to preplan your sorting needs as well. Filing programs sort your data nto alphabetical or numerical order based on selected fields, known as "key fields." Generally, the key field can be changed whenever you like, but in some programs you must identify he key field when you first create the file.

VHAT ABOUT 'INTEGRATED SOFTWARE"?

There exist two main methods of sharing nformation between applications. Truly integrated software combines a data base, a word processor, and (usually) a spreadsheet in a single package. Sometimes separate programs are ile-compatible, meaning you can use data rom one program in another. With filing software, "integration" of either kind will often llow you to take a report from a data base and lace it in a word processor.

Because there's so much software to cover, ve have included only one example of truly ntegrated software—*AppleWorks*. Some data bases, such as *PFS:File* and *Bank Street Filer*, ire part of integrated software families; the ndividual elements are sold separately. In either case, trading information across program ines is easy.

VHAT CAN'T A ILING PROGRAM DO?

The borderlines separating filing programs, word processors, and spreadsheets can e fuzzy sometimes. Each has its strong suit, but each also dabbles in the other's capabilities.

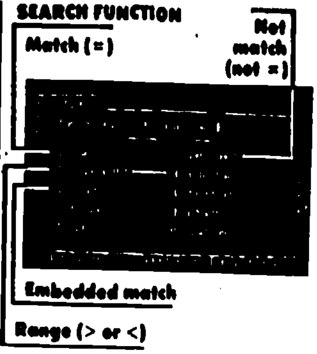
RELATIONAL? WHAT'S THAT?

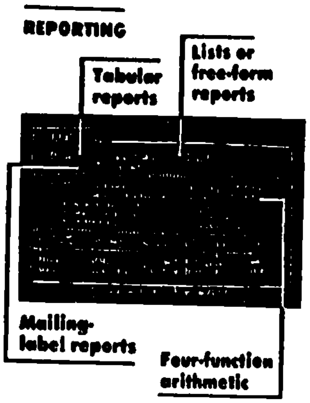
As you cruise your local software store in search of the ultimate data base, you're likely to run into the term "relational"—as in "powerful, versatile relational data-base manager." What is this? A data base to keep track of your relatives?

Not exactly. What makes a data-base program relational is the ability to grab information from more than one file at the same time. Regular filing software can use information from only one file at a time.

For example, a relational package could automatically pull Aunt Jennifer's address from your ADDRESS file, find out what she gave you for your birthday from a separate INCOMING GIFTS file, and then print a report as a perfect thank-you note.

Most relational data bases are quite expensive, although a few, like *PractiBase* from PractiCorp., now cost under $100. However, they still require lots of memory (figure on 256K to start) and are more complicated than a simpler filing program.





Word processors are designed to manipulate text. You can type information, move pieces of text around, and print out in a variety of formats. Most word processors have a search function like a data base, but your ability to organize and reorganize information is limited, and (almost always) you have no mathematical functions.

Spreadsheet software works best with numbers; words are generally inserted solely as labels. They are not usually strong with text manipulation, and the reporting function is limited in format choices.

A data base deals with words and numbers equally well—in either case, the letters or figures represent information to be put into categories. However, the amount of data (text or numbers) you can include in a single record varies widely from data base to data base. In some, you can fit only 24 characters into a single field. Others allow as much as four or five pages of text under a single heading. Still, document-length information often requires a word-processing program.

Similarly, the mathematical abilities of some data bases may be all you need to juggle numbers. But, if you'll be working with a lot of figures, changing some to find out what effect the change will have on others, then a spreadsheet is your tool of choice.

A QUESTION OF STYLE

Everyone deals with information in distinctive ways. Some well-organized souls have every name, date, and number carefully tucked away. Most of us "know it's here someplace" and rummage furiously for the crucial scrap of paper when we need it. A computer filing program makes sense, no matter which group you belong to. The organized individual achieves a loftier state of control, while the harried scrap-hunter enters a mound of data into the computer and lets the software sort it out. FC

STUDENT ACTIVITIES:

You are a member of the staff of the Department of State. You have the task of analyzing social and economic indicators for countries in The Middle East, Latin America and Africa. The purpose of your analysis is to be prepared to make recommendations to your superiors on foreign policy decisions affecting these regions. This is an ongoing task. From time to time you are expected to make specific recommendations regarding foriegn aid, national defense, foreign policy statements, legislation, diplomatic initiatives and responses. Below are projects you have been asked to complete within the next few days.

Use the data file to complete these projects.

PROJECT 1

Answer the following questions. The answers will be used in briefings for key senators and foreign service personnel.

1. How many countries are in Africa?

2. How many countries are in Central America?

3. How many countries are in South America?

4. How many countries are in the Carribean region?

5. How many countries are in the region referred to as The Middle East?

6. Which African Nation had the the highest infant mortality rate in 1961? 1981?

7. Which South American nation had the highest infant mortality rate in 1960? 1981?

8. Which country in each of the following regions had the highest literacy rate in 1980? (North America, Central America, South America The Middle East)

9. Which country in all five regions had the longest life expectancy?

10. Which nations in Africa and Central America had an average annual population growth rate of more than 3.0 percent between 1960 and 1961?

11. Which nations in Central America had more than 50 percent of the labor force engaged in agriculture in 1980?

12. Which African nations had more than 50 percent of the labor force engaged in agriculture in 1980?

13. Which African nation had a population of 32 million in 1981?

14. Which nation in the entire file had the largest population in 1981?

15. Which nation in South America had the highest population per physican in 1980? Lowest?

16. Which nation in Africa had the highest population per physican in 1980? Lowest?

17. Which nations in North America, Central America, South America, The Middle East and North Africa, And Sub-Saharan Africa have a gross national product per capita of more than $10,000.00 in 1981?

PROJECT 2

The United States is considering an increase aid in those categories designed to improve health care. Determine which countries in South America are eligible for this type of aid. Recommend three countries for consideration. When determining which countries to recommend analyze life expectancy, infant mortality and population per physican. Use these factors to choose the three countries you will recommend. Indicate the location of each country selected on a blank outline map. Write several paragraphs describing why each country was selected. Attach printouts of data used in making your decision to the maps and written description.

PROJECT 3

This assignment is a request for information that will help the Secretary of State decide which African countries to approach with incentives to become trading partners. Countries with a high per capita gross national product, above average literacy rates for their region, higher than average exports and lower than average percentage of the labor force involved in agriculture should be considered. Use the data file to narrow your choices and recommend a total of five African countries. Show the location of your choices on a blank outline map and write several paragraphs defending your choices. Attach the printouts used to narrow your choices to the map and narrative.

```
COUNTRY   REGION        POP/PHY LIFE EXP.1981 INFANT MOR 1981 POP 1981
--------- ------------- ------- ------------- --------------- --------
Colombia  South America 1,920   63                55              26.4
Chile     South America 1,920   68                42              11.3
Bolivia   South America 1,850   51               129               5.7
Paraguay  South America 1,710   65                46               3.1
Brazil    South America 1,700   64                75             120.5
Ecuador   South America 1,620   62                80               8.6
Peru      South America 1,390   58                85              17.0
Panama    South America   980   71                21               1.9
Venezuela South America   950   68                40              15.4
Uruguay   South America   540   71                39               2.9
Argentina South America   530   71                44              28.2

=====AVERAGES========11 RECORDS========================================
                        1,373.6364.73             59.64             21.91
```

```
COUNTRY             REGION GNP 1981
------------------- ------ --------
Libya               Africa 8,450
South Africa        Africa 2,770
Algeria             Africa 2,140
Tunisia             Africa 1,420
Ivory Coast         Africa 1,200
Congo               Africa 1,110
Cameroon            Africa   880
Zimbabwe            Africa   870
Nigeria             Africa   870
Morocco             Africa   860
Egypt               Africa   650
Zambia              Africa   600
Lesotho             Africa   540
Liberia             Africa   520
Mauritania          Africa   460
Senegal             Africa   430
Kenya               Africa   420
Ghana               Africa   400
Sudan               Africa   380
Togo                Africa   360
Madagascar          Africa   330
Niger               Africa   330
Central Africa Rep. Africa   320
Benin               Africa   320
Sierra Leone        Africa   320
Guinea              Africa   300
Tanzania            Africa   280
Somalia             Africa   280
Rwanda              Africa   250
Burkina Faso        Africa   240
Burundi             Africa   230
Uganda              Africa   220
Zaire               Africa   210
Malawi              Africa   200
Mali                Africa   190
Ethiopia            Africa   140
Chad                Africa   110
Mozambique          Africa     0
Angola              Africa     0

=====AVERAGES=======39 RECORDS======
                           759.49
```

```
COUNTRY              REGION % LITERACY 1980
-------------------- ------ ---------------
Tanzania             Africa 79
Zimbabwe             Africa 69
Tunisia              Africa 62
Somalia              Africa 60
Zaire                Africa 55
Uganda               Africa 52
Lesotho              Africa 52
Madagascar           Africa 50
Rwanda               Africa 50
Kenya                Africa 47
Zambia               Africa 44
Egypt                Africa 44
Ivory Coast          Africa 35
Algeria              Africa 35
Nigeria              Africa 34
Central Africa Rep.  Africa 33
Mozambique           Africa 33
Sudan                Africa 32
Benin                Africa 28
Morocco              Africa 28
Malawi               Africa 25
Burundi              Africa 25
Liberia              Africa 25
Guinea               Africa 20
Togo                 Africa 18
Mauritania           Africa 17
Chad                 Africa 15
Sierra Leone         Africa 15
Ethiopia             Africa 15
Mali                 Africa 10
Senegal              Africa 10
Niger                Africa 10
Burkina Faso         Africa  5
South Africa         Africa  0
Cameroon             Africa  0
Congo                Africa  0
Libya                Africa  0
Angola               Africa  0
Ghana                Africa  0

====AVERAGES=======39 RECORDS=========
                          29.03
```

```
COUNTRY             REGION % AGRI. LABOR
------------------- ------ -------------
Niger               Africa 91
Rwanda              Africa 91
Central Afri   R.p. Africa 88
Madagascar          Africa 87
Lesotho             Africa 87
Malawi              Africa 86
Chad                Africa 85
Burundi             Africa 84
Tanzania            Africa 83
Uganda              Africa 83
Cameroon            Africa 83
Guinea              Africa 82
Burkina Faso        Africa 82
Somalia             Africa 82
Ethiopia            Africa 80
Ivory Coast         Africa 79
Kenya               Africa 78
Senegal             Africa 77
Zaire               Africa 75
Mali                Africa 73
Sudan               Africa 72
Liberia             Africa 70
Mauritania          Africa 69
Zambia              Africa 67
Togo                Africa 67
Mozambique          Africa 66
Sierra Leone        Africa 65
Zimbabwe            Africa 60
Angola              Africa 59
Nigeria             Africa 54
Ghana               Africa 53
Morocco             Africa 52
Egypt               Africa 50
Benin               Africa 46
Tunisia             Africa 35
Congo               Africa 34
South Africa        Africa 30
Algeria             Africa 25
Libya               Africa 19

=====AVERAGES========39 RECORDS=========
                           67.92
```

```
COUNTRY             REGION EXPORTS
------------------- ------ -------
South Africa        Africa 22,670
Nigeria             Africa 18,727
Libya               Africa 16,391
Algeria             Africa 14,056
Egypt               Africa  3,233
Ivory Coast         Africa  2,586
Morocco             Africa  2,242
Tunisia             Africa  2,209
Angola              Africa  1,744
Kenya               Africa  1,144
Cameroon            Africa  1,079
Zambia              Africa  1,044
Congo               Africa  1,040
Ghana               Africa    878
Zimbabwe            Africa    663
Zaire               Africa    662
Sudan               Africa    658
Tanzania            Africa    566
Liberia             Africa    531
Mozambique          Africa    457
Guinea              Africa    428
Senegal             Africa    416
Ethiopia            Africa    374
Togo                Africa    344
Madagascar          Africa    335
Uganda              Africa    317
Niger               Africa    297
Malawi              Africa    284
Sierra Leone        Africa    277
Mauritania          Africa    259
Somalia             Africa    200
Mali                Africa    154
Rwanda              Africa    147
Chad                Africa    141
Central Africa Rep. Africa    136
Burkina Faso        Africa     75
Burundi             Africa     71
Benin               Africa     36
Lesotho             Africa      0

=====AVERAGES=======39 RECORDS======
                           2,483.7
```

FIELDS IN FILE

COUNTRY:	REGION:
LIFE EXP.1960:	LIFE EXP.1981:
INFANT MOR 1960:	INFANT MOR 1981:
POP/PHY:	% LITERACY 1960:
% LITERACY 1980:	POP 1981:
% ANNUAL GROWTH:	PROJECTED POP:
GNP 1981:	% AGRI. LABOR:
EXPORTS:	

DEFINITION OF FIELDS

COUNTRY:	Name of Country
REGION:	Geographic region
LIFE EXP. 1960:	Life expectancy at birth in years
LIFE EXP. 1981:	Life expectancy at birth in years
INFANT MOR 1960:	Infant mortality rate per 1,000 live births
INFANT MOR 1981:	Infant mortality rate per 1,000 live births
POP/PHY:	Population per physician
% LITERACY 1960:	Adult literacy rate in percent
% LITERACY 1980:	Adult literacy rate in percent
POP 1981:	Population in millions during mid 1981
% ANNUAL GROWTH:	Average annual population growth rate in percent between 1960 and 1981
PROJECTED POP:	Projected population in millions by the year 2000
GNP 1981:	Gross National Product per capita in dollars in 1981
% AGRI LABOR:	Percentage of labor force in agriculture in 1980
EXPORTS:	Merchandise exports in millions of dollars in 1981

APPENDIX

4

ERIC CLEARINGHOUSE
ON LANGUAGES AND
LINGUISTICS

Q&A

Microcomputers and Second Language Teaching

Prepared by John S. Harrison

What Do We Know about the Impact of Microcomputers on the Second Language Classroom?

Not much. Although large (so-called *mainframe*) computers have been used in the second language classroom for at least the past decade, and there is a body of research documenting their effectiveness, there are apparently no research reports on the use of *microcomputers* in second language instruction and few such reports on their use in other subject areas. However, much can be learned from research on large computers in the schools.

A study by the Educational Research Service (ERS) (1982) is instructive. ERS conducted a nationwide survey on how schools are using computers. Among other things, the 1,484 school districts which responded reported the following:

- Computers (in-house or through contracted services) are used by 91.5% of the respondents; 74.6% of the total respondents use them for both administrative and instructional purposes.
- Major benefits noted were "decrease in time spent on routine tasks" (66%); "information for planning and dissemination—available more quickly" (54.7%); and "new functions not previously possible within budget/personnel constraints now being performed" (54%).
- Major problems cited were "staff not adequately trained in using computers" (24.1%); "misunderstanding about the capabilities of computers" (20.5%); and "insufficient/inadequate software available" (20.2%).
- Factors judged most important to the successful and efficient introduction of computer technology into a school district were "technical training of staff" (43.5%); "availability of software packages" (37.9%); and "staff acceptance of computer technology" (29.3%).

Gerald Bracey's excellent article, "Computers in Education: What the Research Shows," gives specific insight into the effectiveness of instruction via computer. Dividing his observations into achievement outcomes, affective/motivational outcomes, and social outcomes, Bracy notes that "In general, students learn more, retain more, or learn the same amount faster using computers. Unfortunately, no studies have been completed yet that tell us *why* that may be. Achievement gains aside, students often find computers more 'human'—more patient, less critical—than humans." Bracey's remarks are based on two major studies: a meta-analysis of 51 research studies done by James Kulik and others at the University of Michigan (to be reported in detail in the *Journal of Educational Research* in early 1983) and a longitudinal study conducted by the Educational Testing Service in Los Angeles elementary schools over a period of four years.

With regard to studies specifically related to computer use in second language instruction, Holmes and Kidd (1982) give a succinct overview of this effort, and further details are available in Anastasia Wang's (1978) compendium. Olsen's (1980) extensive survey of colleges and universities is also a very useful resource. However, most of these reports, again, center on the use of mainframe computers.

Although research regarding microcomputers and second language instruction is yet to come, one can still develop an awareness of the microcomputer's inherent possibilities and limitations. Novices are well advised to consult such recently developed publications as *The Computing Teacher, Classroom Computer News,* and *Electronic Learning* for information of this kind.

What Skills Do I Need to Use a Microcomputer?

At this point, everyone needs to know how to turn on the computer, load software programs, and manipulate the programs. Until voice recognition by the computer reaches a more advanced stage of development, some degree of typing skill will continue to be important. Being an intelligent computer user would also include familiarity with relevant issues such as privacy, piracy, and information security.

Center for Applied Linguistics
3520 Prospect Street, N.W.,
Washington, D.C. 20007

A question often posed by teachers is, "Do I need to know how to write programs?" Those who answer *yes* see as essential the ability to make the computer do what one wants it to do instead of being at the mercy of the person who has written the program. However, some special types of programs (usually referred to as "authoring systems") allow teachers to easily enter their own material into existing program frameworks. The proliferation of such authoring programs may eliminate the need for most teachers to learn programming.

One thing is clear: the need for teacher training in *using* computers is paramount. Those who want to learn about microcomputers will find a wide variety of inservice courses and training workshops, many tailored for the specific interests of second language teachers. Some language teachers will find that learning a programming language, with its built-in vocabulary and syntax, is analagous to learning another foreign language, although a much easier process. Many community colleges offer good introductory programming courses which will help a teacher decide whether or not to pursue further development of this skill.

What Kinds of Software Exist?

Educational software, often called *courseware,* ranges in scope and complexity from short, stand-alone programs teaching a single concept to sets of programs comprising a complete instructional sequence. Courseware can provide Computer Managed Instruction (CMI) or Computer Assisted Instruction (CAI). The term CMI describes administrative kinds of software such as programs that use the computer to store, analyze, and retrieve data on student achievement. In such programs, the student may take tests directly on the computer or on paper with the results being put into the student's computer record. CAI programs are, indeed, "instructional," and may involve one or more of these formats (Smith 1983):

- *Drill and Practice* presents problems or questions to reinforce specific skills or concepts on which the student has already received instruction. Among other things, these may be games in which students compete against the computer or each other.

- *Tutorial Dialogs* present instruction alternated with questions about the material presented, engaging the student in a kind of dialog to check comprehension.

- *Simulations* set up a model of a scientific or social event. The student interacts with this model by making decisions and subsequently learns the consequences of these decisions.

- *Problem-Solving* programs require calculation of complex formulas or arithmetic expressions. They allow the student to study mathematical topics which require rigorous calculations, to study history and sociology from a statistical perspective, and to perform analyses on data collected in the science laboratory.

Other types of software which may be used for instructional purposes include:

- Programs for word processing, business management, music composition, graphic design, or for connecting equipment such as microcomputers and videotape equipment.

- Database programs for the storage, manipulation, and retrieval of related information.

- "Utility" programs to allow teachers to keep gradebook records or generate printed worksheets and tests.

Of the approximately 100 microcomputer programs in foreign languages with which this author is familiar, about 90% fall into the drill-and-practice category. Since a large percentage of the work involved in learning a second language, especially at the lower levels, concentrates on rote work, this preponderance of drill materials is not necessarily a drawback. Many of the drill programs also incorporate some semblance of tutorial activities. Only a few of the commercially available programs—such as *Mystery House* (French version), published by Sierra On-Line Computing—could be called a simulation.

How Can I Get Under Way?

All practitioners agree that, in theory, the first three steps are to determine your objectives, choose the courseware, and select the microcomputer which will run that courseware. However, to set realistic objectives, the teacher must know what kind of courseware is available, what topics it deals with and at what level, what format it has, and whether it can be satisfactorily manipulated by the students. Once this information is obtained, the question becomes: Does this material help students attain goals that fit into the course?

Three published sources can assist the second language teacher in locating and evaluating courseware for foreign languages. Harrison's (1983) article in *Newsletter 13* of the Northeast Conference on the Teaching of Foreign Languages is the first such effort. Two others are available as a result of federal funding. In the summer of 1982, an institute sponsored by the National Endowment for the Humanities brought together approximately 30 foreign language teachers at the University of Delaware, who, among other activities, compiled *Foreign Language Teaching Programs for Microcomputers: A Volume of Reviews.* Under the aegis of the federal Office of Educational Research and Improvements, the University of Iowa has developed a similar volume entitled *Needs and Development Opportunities for Educational Computer Software for Foreign Language Instruction in Schools.*

To a great extent, the courseware chosen will dictate the brand of microcomputer to purchase, since the courseware for one brand of computer is, in general, not readily transferable to another. However, this situation is changing rapidly. In a report on the West Coast Computer Faire held in March 1983, Erik Sandberg-Diment (1983) reports the appearance of "a number of add-on circuit boards that can be slipped into an IBM PC to change its personality so it will think like an Apple, which means the IBM can suddenly run all of Apple's extensive software library."

Once objectives are set and courseware and microcomputer are acquired, the next consideration becomes how best to use the new technology. The state of Minnesota, a leader in the development of instructional computing, provides useful implementation and training checklists in its manual, *The Use of a Computer to Help Teach the School Curriculum*. Perhaps the trickiest problem to resolve is that most second language teachers are faced with one microcomputer for 30 or more students. Phillips (1983) suggests four possible approaches:

1. Total class instruction using one computer and a large monitor. This works especially well with simulations.

2. Timed-use relay. (The whole class is organized into a variety of small group activities, one of which is using the computer.)

3. Block-time format for independent work. (This is used primarily in self-contained classrooms. Each student is assigned a block of time weekly.)

4. Nonscheduled format.

As microcomputers receive wider acceptance and as budgets permit, one can expect the establishment of laboratory facilities equipped with a number of microcomputers. The technology for such installations already exists. Apple's Schoolbus system, for example, allows a teacher to use current Apple IIs, printers, and disks to form a network of disk-sharing computers. Thirty Apple computers can use programs stored at the instructor's station, eliminating the need for program disks and disk drives at each student station. With Radio Shack's Network 3 Controller, up to 16 Model III microcomputers can be connected to a single host system. Individual students can choose and work with any lesson stored in the host computer. As lessons that include student record-keeping are completed, performance information is automatically stored in the host computer for later review by the instructor.

What Else Can a Microcomputer Do for Me?

Up to now, discussion has centered on the microcomputer as an autonomous unit in the classroom. In fact, the microcomputer can also provide access to materials beyond the classroom, to whole libraries of information stored on mainframes or other microcomputers. These libraries, called databases, can be connected to the microcomputer via telephone. When looking for significant research or for an appropriate exercise for Monday morning classes, databases like that of the Educational Resources Information Center (ERIC) can be tapped from the comfort of the faculty lounge or one's own home. While ERIC is the largest education database, there are many others of interest to foreign and second language educators—Bilingual Education Bibliographic Abstracts, Psychological Abstracts, School Practices Information File, Resources in Computer Education, Resources in Vocational Education, and Exceptional Child Education Resources, to name a few. In fact, Markoff and Shea (1983) note that "Keeping track of all the on-line databases that have proliferated over the past decade requires a database itself." They also list the major database services such as BRS After Dark (significantly lower rates in the evening), the Source (which includes a bank of foreign language lessons), and CompuServe.

What More Is on the Horizon?

The microcomputer already permits significant instruction and practice in reading and writing skills and holds promise for being useful in learning speaking and listening skills. The cutting edge in technological development is in voice synthesis and voice recognition. There already exists a thumbnail-sized computer chip which can produce all the phonemes of human speech, permitting inexpensive text-to-speech synthesis. One can purchase a clock radio (Telestar) which announces the time in English, Spanish, Mandarin, Cantonese, or Taiwanese with excellent intelligibility. Second language teachers at the U.S. Air Force Academy are experimenting with interconnecting the microcomputer and videotape and videodisc equipment. Scott Instruments is already marketing its Voice-Based Learning System (VBLS) which, according to its publicity, "was designed for use by non-computer oriented persons, and requires no programming skills. Instructional lessons are vocally entered in the language to be used by the student. VBLS can understand any language, including utterances resulting from speech impediments. This makes it ideal for bilingual and second language study." Write a letter in one language on the computer and have it printed in another? One American company—Ditronics—is working on this problem. Speak to the computer in one language while a computer in a distant country translates? This is a current project of Japan's Fijitsu Company.

These are but a few of the many innovations in the computer industry which can be adapted to second language learning needs. These and other rapidly-occurring developments hold enormous promise for the near future.

References

Bracey, Gerald. "Computers in Education: What the Research Shows." *Electronic Learning;* (November/ December 1983) 51-54.

Culley, Gerald and George Mulford, eds. 1983. *Foreign Language Teaching Programs for Microcomputers: A Volume of Reviews.* Wilmington: University of Delaware.

Harrison, John. 1982. *Applications of Computer Technology in Foreign Language Teaching and Learning.* (ERIC Document Reproduction Service No. ED 216 552).

———. "Foreign Language Computer Software: What? Where? How Good?" *Newsletter 13* (Northeast Conference on the Teaching of Foreign Languages); (February 1983) 26-30.

Hawkins, William. "Talking Appliances." *Popular Science;* (September 1982) 74-75.

Holmes, Glyn and Marilyn Kidd. "Second Language Learning and Computers." *Canadian Modern Language Review;* 38 (Spring 1982) 503-516.

McComb, Gordon. "Speech, Speech!" *Creative Computing;* (December 1982) 120, 130.

Markoff, John and Tom Shea. "Information Utilities." *Infoworld;* 5 (March 28, 1983).

Needs and Development Opportunities for Educational Computer Software for Foreign Language Instruction in Schools. 1983. Iowa City: University of Iowa.

Olsen, Solveig. "Foreign Language Departments and Computer-Assisted Instruction: A Survey." *Modern Language Journal;* 64 (Fall 1980) 341-349.

Phillips, Wayne. "How to Manage Effectively with Twenty-Five Students and One Computer." *Computing Teacher;* (March 1983) 32.

Sandberg-Diment, Eric. "Fringe Element, Software 'Rentals' Come to Computer Show." *Baltimore Sun* (New York Times News Service), March 30, 1983, C10.

School District Uses of Computer Technology. 1982. Arlington, Virginia: Educational Research Service.

Smith, James, ed. 1983. *Criteria for Evaluating and Selecting Microcomputer Courseware.* Baltimore: Maryland State Department of Education.

The Use of a Computer to Help Teach the School Curriculum. 1982. Minneapolis: Minnesota Association for Supervision and Curriculum Development, Minnesota Association of School Administrators, and the Minnesota Department of Education.

Wang, Anastasia, ed. *Index to Computer-Based Learning.* 1978. Milwaukee, WI: University of Wisconsin, Instructional Media Lab.

Database Services

BRS, 1200 Route 7, Latham, NY 12110; (800) 833-4707, (518) 783-1161.

CompuServe, 5000 Arlington Centre Blvd. P.O. Box 20212, Columbus, OH 43220; (614) 457-8600.

DIALOG Information Retrieval Service, 3460 Hillview Ave., Palo Alto, CA 94304; (800) 227-1960, in California (800) 982-5838.

This publication was prepared with funding from the National Institute of Education, U.S. Department of Education under contract no. 400-82-0009. The opinions expressed in this report do not necessarily reflect the positions or policies of NIE or ED.

THREE MAJOR APPROACHES TO DEVELOPING COMPUTER-ASSISTED LANGUAGE LEARNING MATERIALS FOR MICROCOMPUTERS

David H. Wyatt

ABSTRACT

Whenever a new project to produce computer-assisted learning materials is begun, one of the most fundamental decisions concerns the selection of an approach to program development. There is a choice between at least three major approaches: using a general purpose programming language, an educational programming language, or an educational authoring system. There are considerable differences in factors such as programming expertise and development time required by the three approaches. Perhaps the most dramatic example of this is the fact that the third approach demands no programming expertise whatever. It is therefore important to be aware of the alternatives in order to make an informed choice of the most suitable approach for each materials development project.

Questions concerning the 'best' approach to developing courseware have often been the subject of intense debate (Holmes 1983; Wyatt 1983a). Some have argued that the current strong interest in producing one's own materials is just a short-term phenomenon and that in the future CALL materials will largely be purchased in ready-to-use form. In this view, considerations regarding the optimum method of developing courseware will soon be relevant only to a small number of professionals. Others maintain that the role of language teachers (and other subject specialists) in software development should be limited to specifying the instructional content and its style of presentation on the screen; all questions of how to realize the material in programming terms are to be left to programmers (Bork, 1981).

Options

In spite of these views, the fact remains that an increasing number of language teachers and curriculum developers are becoming interested in producing their own courseware. This article is addressed to them in an attempt to provide a guide to the various options which should be considered. Three major vehicles for creating courseware will be examined and compared in some detail: general-purpose programming languages, educational programming languages, and educational authoring systems. The first two of these require the user to acquire programming skills, while the third type demands only the simplest level of computer literacy. Perhaps the most important point to bear in mind before we begin to focus on the different approaches is that *all* of them can be successfully used in different situations. They all have weaknesses as well as strengths, and the choice between the approaches depends very much on your individual circumstances and requirements.

To facilitate the comparison of development methods, we will consider CALL software as essentially being composed of three elements: the language content, the driver, and the management system. This is an oversimplification, and is not equally appropriate for all types of educational software, but it will serve to underline some fundamental points. The *content* may be separate from the program, residing in data files, or may be incorporated directly into the program itself. The *driver* is the computer program which takes the content and manipulates it so as to present examples and questions, accept answers, and provide appropriate responses. In general, a different type of CALL exercise or activity requires a different (and usually a separate) driver program. The *management* system is the element which determines how students can enter and use the courseware, what route they follow through the materials, and what score and progress information is recorded. Management systems usually involve some separate programs, but

David H. Wyatt is currently director of an educational consulting firm in computer-assisted language learning. Educated at Cambridge University and the American University, he is a consultant to the Center for Applied Linguistics and the National Clearinghouse on Bilingual Education. His former positions include Director of CAI at a large system of ESL schools management elements are generally also built into each driver program.

General-Purpose Programming Languages

The first main approach to the development of courseware involves the use of general-purpose programming languages such as BASIC and Pascal. In practice, BASIC has been the overwhelming choice of producers of commercial educational software for microcomputers. It has been estimated that 80 per cent of the software to date has been written in BASIC and the remainder in machine language with no significant contribution from any of the other programming languages.

There are excellent reasons for the popularity of BASIC in the creation of software for microcomputers. In theoretical terms, programmers working in general-purpose languages are in much more direct contact with the computer's microprocessor and memory than with the other two approaches. It is possible to exert much more control, and more flexible control, over each step of the operation of the computer. Programmers may design and create every step of the main drivers for their exercises. Similarly, they have complete control over the management system elements in their courseware. For example, the other two approaches frequently offer either rudimentary management systems or demand the presence of a second disk drive if score recording and other management capabilities are to be used. For these reasons, one project directed by the author (Wyatt 1983b) made use of BASIC to create a powerful management component which required only one disk drive for full utilization.

Practicality also dictates strong arguments in favor of general-purpose languages. One important factor is the lack of dependence of the educational programmer on other factors. In one well-known recent case, for instance, a new version of an educational programming language was released with the promise of a management component soon to follow. However, the management component was delayed and became available much later than expected, which must have caused difficulties for those to whom the management capabilities were important. A second practical point concerns the position when new microcomputers are released. In virtually every case, a version of BASIC is immediately available for new microcomputers, whereas a period of months or even years may pass before effective educational programming languages or authoring systems are produced for the computer.

Since general-purpose languages were intended for a wide range of different applications, these languages have few if any commands oriented towards the needs of the educator.

However, the reader has probably already perceived the roots of a problem during this discussion of general-purpose languages—control over all the individual steps of an exercise program, desirable as it may be, implies the need for time-consuming specification of all of those steps. In general, the time required to create a given student activity or exercise in a general-purpose language is significatly greater than with the other two approaches. To be more specific, it is the programming of the new driver (and, to a lesser extent, inclusion of the management system elements) which demands the greater investment of time. A second disadvantage is the considerable learning time required for the novice to become sufficiently expert in a general-purpose language in order to produce moderately sophisticated educational programs.

One of the main reasons for the time-consuming nature of producing educational programs in general-purpose languages is their lack of educational 'power.' Since general-purpose languages were intended for a wide range of different applications, these languages have few if any commands oriented towards the needs of the educator. After only a little experience with general-purpose languages, even novices in educational programming begin to recognize the lack of convenient commands. At a lower level, desirable features of a programming language would include simple commands to cause the program to wait a specified number of seconds and also accept student input in a fully controlled manner (filtering out any unwanted or troublesome keypresses). At a higher level, we would look for some ability to perform natural language answer processing—to search students' answers for key morphemes, words, or phrases (Pusack 1983). Ideally, we would prefer intelligent answer processing, in which students' wrong answers are compared with the expected response, 'marked up' to indicate problem letters or words, and then turned back over to students in an 'edit' mode so that only the mistakes need be retyped at the second attempt. Even the lower-level educational commands described above are entirely absent from BASIC, although they can be duplicated by the programmer with varying degrees of difficulty and effort. These added capabilities are generally developed in the program in the form of subroutines which can be reused whenever they are needed in this or future driver programs. General-purpose language programmers tend to build up large libraries of such subroutines to facilitate their developmental work. Producing the higher-level answer processing features is a major undertaking, however, and BASIC is in any case not suitable for the final form of such routines because of technical reasons (for one thing, it runs too slowly).

Serious as the deficiencies of a general-purpose language may appear, there are effective remedies for many of them. In some cases, it is possible to buy a ready-made set of educational subroutines which will provide the low-level capabilities described above. As one example, the Minnesota Educational Computing Corporation has produced a well-documented collection of subroutines in BASIC for the Apple II (MECC 1980). These ready-made routines can immediately be used in creating courseware and provide the beginning programmer with a head start in general-purpose language work. In at least one instance, an enhancement to BASIC is available which will provide an excellent high-level answer processing capability (Tenczar *et al* 1983). When the problems are solved in this way, the advantages of intimate contact with the computer and complete control over program operation become very compelling.

Educational Programming Languages

The second major type of approach to developing courseware involves the use of programming languages specifically designed to meet the needs of educators. The best known of these educational programming languages in the context of microcomputers is PILOT (Burke 1983), although a recent development known as EnBASIC is also of considerable interest (Tenczar 1983). Because of their specific orientation, these languages incorporate a range of very convenient commands which provide both trivial and powerful educational facilities for the programmer

We will illustrate the range of new commands by reference to the low- and higher-level facilities discussed in the previous section, using the SuperPILOT version of PILOT for the Apple II as our example. At the lower level, PILOT provides convenient commands to cause the exercise to wait a specified number of seconds before proceeding and to 'bombproof' the keyboard so that only desirable, meaningful keypresses have any effect on the screen. In both these cases, simple one-line PILOT commands can permit the program to ignore unwanted keypresses. At a higher level easy-to-use facilities also provide powerful answer processing. One example is the PILOT 'key search' capability. Using this, students may be permitted to enter their answer in a relatively free manner. Once entered, their input can be searched for the significant part of the answer, which might be an affix, single word, phrase, etc. Less important parts of their input can be disregarded, so that spelling mistakes can be tolerated where appropriate instead of causing an otherwise acceptable answer to be judged incorrect. However, PILOT stops short of providing highly intelligent answer processing. EnBASIC is the only current programming language for microcomputers in which this is available. Using EnBASIC, students' answers can be compared against the predicted responses, and then automatically 'marked up' with a simplified set of proofreader's symbols to indicate problem letters and words. Students are then put into a special 'second chance' mode in which they edit their original answer, changing only the incorrect portions. (EnBASIC also provides most of the facilities of PILOT mentioned above.)

In reality, time-saving is probably no longer the most important advantage of educational programming languages.

SuperPILOT includes a number of other features in a convenient package to take advantage of specific capabilities of the Apple II microcomputer. These include utilities to permit creation of special characters and diacritics (such as those needed in foreign language courseware), simple music and sound effects, color graphics, and control over external videotape and videodisc players. At least one other version of PILOT for the Apple, PILOT Plus, provides very similar capabilities including provision for touch-sensitive screen input, and other versions of PILOT now exist for an increasing number of microcomputers. It should be stressed again that programmers in a general-purpose language can supply themselves with very similar capabilities. Features such as special character fonts, sound and musical effects, color graphics, and control of video devices can either be developed by the programmer as reusable subroutines or purchased commercially as 'utilities' in ready-to-use form. This piecemeal solution lacks the convenience of the package provided by an educational programming language, but this deficiency must be balanced against the greater control and flexibility involved in being able to select particular utilities and subroutines rather than being limited to the single packaged facility

The powerful, convenient features described above make it potentially quicker and easier to program a new exercise driver in an educational programming language. A further saving of time is posible if a management system is also available as part of the package, as is the case with the PILOT Log component of SuperPILOT. The time-saving aspect of educational programming languages should not be overstated. Programming time will be less than in general-purpose languages, but will nevertheless be of the same order of magnitude. It is generally agreed that the time taken for a beginner to become proficient will also be significantly shorter in an educational programming language, but is is again important not to overestimate the gain. Claims of a few hours' learning time for educational programming languages refer only to the use of the simplest commands. In order to use the capabilities of the language at a moderately sophisticated level, far more time is required. In its original form, PILOT was a greatly simplified language, but the extended versions produced for microcomputers have added more complex commands and capabilities. Some have argued that these extended PILOT languages actually require the same learning time as general-purpose languages (Hardy and Elfner 1982). In reality, time-saving is probably no longer the most important advantage of educational programming languages. Perhaps their chief benefit lies in high-level capabilities such as natural language answer processing which enable programmers to develop courseware which is significantly more open-ended and sophisticated than would otherwise be possible.

In this generally positive situation, however, there is the potential for some serious problems. Educational programming languages represent an attempt by their designers to predict what types of commands and capabilities educators will need. Since computer memory capacities are limited, programming languages tend to represent compromises—not all the desirable commands and features can be incorporated, and inclusion of a range of new educational commands may imply the loss of some of the facilities available in general-purpose languages. Indeed, with the original version of PILOT, the range of commands was deliberately kept to a minimum to permit rapid learning of its capabilities (Merrill 1982). In any case, the tendency to reduce the scope of commands usually available in general-purpose languages may lead to problems if the designer of the educational language has not fully anticipated the needs of the courseware developer.

We will illustrate the potential for difficulties with reference again to different versions of PILOT for microcomputers. In one early version, there was relatively little memory space allotted for the textual content of the exercise, so that repeated accessing of the diskette was necessary in order to load and display the necessary screen contents. This caused frequent short delays in the running of many language-oriented exercises. Apparently the designers of this PILOT version had not considered the delays to be significant, and some educators also found them acceptable. A sizable number of teachers, however, felt that the delays were irritating enough to students to seriously jeopardize the effectiveness of their courseware. This problem has apparently been much reduced or entirely eliminated in recent versions of PILOT for microcomputers.

A second specific instance of a problem not anticipated by the educational language designer concerns the character set editor available with PILOT. In general, this is a powerful utility which permits the creation of new symbols and letters needed for foreign language instruction. Unfortunately, however, in PILOT it is necessary to designate a specific key on the keyboard whose symbol will be replaced by the new character. Thus, the additional forms of the vowels in French (acute, grave, etc.) cannot be represented in any very logical fashion; ê, â, and ô must be represented by quite different keys. Using a general-purpose language, however, it can be arranged for the circumflex form to be generated simply by pressing the original vowel followed by a single 'circumflex key'—as it happens, there is an appropriate symbol on the keyboard. Similarly, all acute vowels could be generated by pressing the base letter followed by the slant on the keyboard, whereas PILOT would require them to be represented by different symbols which could not be related to the base vowels. This is a complex but apt illustration of the difficulty of trying to predict, as the designer of an educational programming language, what the requirements of the users are likely to be. Finally, a more general observation which has been made regarding educational programming languages such as PILOT is that they are structured so as to be well suited only to tutorial and drill-and-practice types of CALL, lacking the flexibility for use in developing more innovative and communicative activities.

The authoring system ensures that the screen presentation and question flow are handled very professionally.

Most of these drawbacks are theoretically avoidable, however, and as the relatively young field of microcomputer-assisted learning matures it can be seen that educational programming language designers are improving on their early efforts. Certainly one of the newer languages, EnBASIC, does not seem to suffer from any of these problems. It functions as an enhancement to Applesoft BASIC, so that virtually none of the capabilities of that general-purpose language are lost. At the same time, it supplies some very powerful commands and features for educational applications. The only major sacrifice is a sizable part of random access memory—approximately 16,000 bytes are occupied by the language itself.

Educational Authoring Systems

The third general method of developing courseware is through the use of an educational authoring system. Unfortunately, there is often considerable confusion in terminology between educational programming languages and educational authoring systems. The criterion used for the classification in this paper is whether the courseware developer is required to perform any type of programming while using the method; if so, it is at least partly composed of an educational programming language. If no actual programming is required, then the method is a pure authoring system.

This last point embodies the outstanding advantage of authoring systems: teachers do not need to learn any programming language in order to use them successfully. In fact, all that is absolutely required in terms of computer knowledge is familiarity with the keyboard, disk drives, and diskettes—in other words, an elementary level of computer literacy. This means that the time involved in learning to use an authoring system is essentially required for study of the features of the system itself rather than those of a computer or programming language. A dramatic reduction in learning time is thus possible. The simpler authoring systems can actually be mastered in a period of a few hours or less. More powerful authoring systems have features and techniques which may take considerably longer to master, but they still offer a tremendously shorter learning period than the approaches which employ programming languages.

Authoring systems can also greatly accelerate the speed of development of courseware. Once the language content of the lesson has been specified in the instructional design phase—required in any approach to CALL—the authoring system will dramatically reduce the time taken to translate the plans into the form of software. As an example, let us briefly examine the operation of a moderately sophisticated authoring system, the Assisted Instructional Development System, or AIDS (Wolfe 1982). AIDS is largely a menu-driven authoring system. When first activated by inserting the diskette and turning the computer on, the system offers a menu of actions from which to choose: creating a new exercise, editing existing exercises, transferring files from one student diskette to another, deleting unwanted exercises, etc. If 'create a new exercise' is selected, the user is then prompted for the information required virtually every step along the way. First, a student diskette must be inserted on which the new exercise is to be recorded. For each question of the exercise, the developer is asked to type in information such as the text to be read (if any), the question to be asked (if separate from the text), the correct answer or alternatives, comments to the student if correct, and to predicted wrong answers, with specific feedback messages on each of the individual problems involved. As each question is completed, the system automatically stores it on the student diskette previously inserted.

When all the questions for the exercise have been typed in, the system then asks if the developer wishes to review and edit the new exercise immediately. Because the process is guided and structured at every point by the authoring system, the creation of new exercises proceeds rapidly and smoothly. When the resulting material is used by students, the authoring system ensures that the screen presentation and question flow are handled very professionally. This is rather gratifying to the novice user, as even the first attempts appear quite polished when used by students. The system also makes it very difficult to create material which will 'crash' because of programming defects, so that the debugging process is greatly shortened and is essentially limited to correcting errors in the language content.

It can be seen that authoring systems supply all the necessary program drivers in ready-made form. Some systems also provide built-in management capabilities. With the AIDS system, for example, there is an extensive score recording and exercise routing capacity. The system also provides a simple keyword answer processing feature.

Although AIDS is an excellent example of a medium-power authoring system, it is not specifically designed for foreign language applications, as is shown by the absence of ready-made foreign alphabet character sets. An example of an authoring system specifically for foreign language application is the DASHER (Pusack 1982) authoring system, which offers foreign characters and diacritics as well as a number of other interesting features. It incorporates an intelligent answer processing capability which is similar in power to that of EnBASIC. Mistakes in answers are indicated by simple 'markup' symbols, and students are then put into an editing mode which enables them to attempt to correct their answers by changing only the incorrect portions. The power and convenience of this edit mode makes this feature one of the best of its kind among current microcomputer-based systems and languages.

DASHER and AIDS are examples of medium-power authoring systems. At a higher level, a number of systems exist which will provide even more sophisticated capabilities. The PASS system from Bell and Howell, for example, will permit the control of videotape and videodisc players from within the courseware. At a lower level, there are a host of inexpensive authoring systems available. Many of these offer a range of relatively rigid exercise formats—typically including true/false, multiple choice, and matching—designed for general educational purposes. Recently, however, there has been a tendency to offer some language-oriented authoring systems which will generate a very limited but indefinitely reusable type of activity. For example, Clozemaster (Jones 1982) will generate a wide variety of different cloze exercises from any paragraphs which are typed into it.

The disadvantages of the authoring system approach are as obvious as its advantages: such systems are relatively quick and easy to use because they embody a built-in educational methodology and program logic. Almost invariably they are strongly *instructional* in nature, and it would be difficult or impossible to use them to create more open-ended or communicative activities of a *collaborative* type (Wyatt 1983c). In general they are not even suitable for instructional programs of the tutorial type, since their branching capabilities are very limited or non-existent. In the great majority of cases, they are suitable only for the creation of drill-and-practice exercises and quizzes. However, if it is precisely this type of courseware that is desired they offer a highly cost-effective option that deserves very serious consideration.

Summary

In summary, there is no single 'best' method of developing courseware. This paper has presented a wide spectrum of possible approaches to the creation of materials for computer-assisted language learning, all of which can be appropriate in different circumstances and for different prposes. It is important that prospective developers of computer-based curricula be aware of the alternatives open to them since different approaches may offer very significant variations in time and energy required for completion of a given project. It is to be expected that the situation will continue to change as new or improved versions of these approaches become available.

REFERENCES

Bork, Alfred. 1981. *Learning with Computers.* Bedford, MA: Digital Press.

Burke, Robert. 1983. *CAI with PILOT.* Englewood Cliffs, NJ. Prentice-Hall.

Hardy, R. Reed, and Eliot S. Elfner. 1982. Writing courseware on the TRS-80. *Creative Computing.* October 82: 34-35.

Holmes, Glyn. 1983. Creating CAL courseware: some possibilities. *System.* 11 (1).

Jones, Christopher. 1982. *Clozemaster* (booklet and diskette) London: WIDA Software.

Merrill, Paul F. 1982. The case against PILOT *Creative Computing* July 82. 70-77.

Minnesota Educational Computing Consortium 1980. *Programmer's Aid Volume I* (Booklet and diskette) St. Paul, MN: MECC.

Pusack, James P. 1983. Answer-processing and error correction in foreign language CAI. *System* 11 (1).

__________. 1982. *DASHER: A Natural Language Answer Processor.* Iowa City, IA: Conduit.

Tenczar, Paul, Stanley Smith, and Allen Avner. 1983. *A Manual of EnBASIC Usage.* Wentworth. NH: COMPress.

Wolfe, George. 1982. Assisted instruction development system. *BYTE*, August 82. 408-414.

Wyatt, David H. 1983a. Computer-assisted language instruction: present state and future prospects. *System.* 11 (1): 3-11.

__________. 1983b. *Teachers' Management Program* (handbook and diskette). New York. NY: Regents ALA.

__________. (forthcoming 1983c) *Computer-Assisted Learning in ESL.* Washington, DC: Center for Applied Linguistics.

Graphing Functions with a Computer

Mark Harris
Appalachian State University

Introduction.

The purpose of this teleconference is to give you an easy way to graph and compare functions on a microcomputer without the aid of a commercial plotting package, and to show you some uses of computer graphing. You will have at your disposal a short BASIC program for the Apple II called 'GRAPHIT' (listed at the end of this handout). The program is a bare-bones graphing program: it shows pertinent information about the function and its scaling on a main menu, but the actual graph has no explanatory text or 'tic marks'. Since the most frequent use of such a program is for investigating shapes of curves rather than reading detailed numerical data, you will probably not miss the frills which are omitted.

Preparation.

You can run the program by simply inserting the GRAPHIT disk in your drive and turning on the Apple's power, or by entering 'RUN GRAPHIT' if the operating system has been previously 'booted' (powered up) on the machine. You will see the following initial menu:

```
          GRAPHIT

       BY MARK HARRIS

1. DOMAIN: [0,6.2832]
2. RANGE:  [-1,1]
3. F(X)=SIN(X)
4. CLEAR SCREEN
5. SHOW GRAPH
6. PLOT
7. QUIT

          WHICH? (1-7)
```

The following exercises will give you quick tour of the program. In the teleconference and in the follow-up exercises we will explore various uses of GRAPHIT.

1. Press '6' to plot the curve defined by the given data. You will see one cycle of the sine graph along with the axes. View the graph to your satisfaction, then press a key to return to the menu.

2. Press '1' to change the domain. You will be prompted to enter the left endpoint, and the cursor will be positioned over the current value. You can accept the current value by pressing <Ret> or you can enter an new value. (Moving the cursor with the right arrow key is the same as typing the numbers under the

cursor.) In this case just press return.
The next prompt asks for the left endpoint. Enter the number 12.5664 (2 Pi) and press <Pet>. From the main menu select option 6 to plot again. What do you see? Press a key to return to the menu.

3. Although your graph appears to be gone, it still resides comfortably in memory. Press '5' to show the graph, then press another key to return to the menu.

4. With the previous graph still in memory we will add a second curve for comparison. First select option 4 to change 'CLEAR GRAPH' to 'KEEP GRAPH'. Next press '3' to enter a new function; you will see the prompt

```
    F(X)=SIN(X)
```

with the cursor positioned over the 'S'. Key in the function COS(X). (One way to do this is to type COS and then press the right arrow key three times to accept the characters '(X)'.) Press <Ret> to enter the function; if your entry was a legal Applesoft function then you are returned to the menu. If your function was misspelled or otherwise improper, the computer will beep at you and wait for you to retype it. Assuming that you are back at the menu with the function shown as F(X)= COS(X), press '6' to plot. You should now see the graphs of both sin(x) and cos(x) on the interval [0,2 Pi]. Press a key to return to the menu.

5. In the previous examples the default range of [-1,1] was a good choice since it represents the extremes of both the sine and cosine functions. In general the estimation of a reasonable vertical scaling requires an understanding of the function(s) being graphed. One convenient option offered by GRAPHIT ia 'automatic scaling'; you just enter the domain and let the program determine the minimum and maximum values of the function. To illustrate this feature, set the domain to [-2,2], the function to X^2 (the ^ symbol is used for exponentiation), option 4 to 'CLEAR', then press '6' to plot. The parabola is displayed fine from -1 to 1 but the remaining portion of the curve is off the screen. Return to the menu and press '2' to change the range. The screen shows

```
     SCALING = MANUAL
        OK? (Y,N)
```

Press 'N' to change to AUTOMATIC scaling, then press 'Y' to accept. (If you accept manual scaling you are prompted for th low and high values of the range.) Press '6' again to graph the function. This time the Y-axis is scaled from 0 to 4 to show the complete section of the parabola from X = -2 to 2.

This completes the tour of GRAPHIT which will prepare you for the teleconference. You may want to experiment with additional graphs; you can't do any harm!

Follow-up exercises.

1. Graph SIN(X) and SIN(2*X) on the same screen, using the domain [0,6.2832] and range [-1,1]. Experiment with other period changes in trigonometric funtions.

2. Graph SIN(X) and 2*SIN(X) on the same screen, using the domain [0,6.2832] and range [-2,2]. Experiment with other amplitude factors (including negative values).

3. Graph SIN(X) and SIN(2*X-1) on the same screen, using the domain [0,6.2832] and range [-1,1].

4. Graph X^2 and (X-1)^2 on the same screen, using domain [-1,2] and range [0,4].

5. Graph X^2 and X^2 + 1 on the same screen, using domain [-1,1] and range [0,5].

6. Graph X^2 and X^2 -2*X + 2 on the same screen, using domain [-1,2] and range [0,5].

   Problems 4 - 6 illustrate how graphs are shifted vertically and horizontally. The more general problem is to graph F(X-A) + B if you know what the graph of F(X) looks like. Experiment with graphing F(X) and F(X-A) + B on the same graph for various functions F(X).

7. Find all the zeros of F(X) = X^3 - 2^2 - X + 1 to within .1 units. Here is one approach: start by graphing the function on the domain [-3,3] with automatic scaling to get a general idea of the location of the roots. If you think you see a root in the interval (say) [0,1], graph again on this smaller interval. If the graph still crosses the X-axis in the new interval, use a still smaller interval and graph again. If (say) you graph on the interval [.4,.6] and you still see a crossing then it is safe to say that one of the zeros equals .5 to within .1 units.

8. Graph F(X) = 1/(X-1) on the domain [-2,2] with automatic scaling. Not a pretty picture, is it? Change the scaling to manual, enter the range [-5,5], and graph again. Better, right? Can you explain what went wrong the first time? Think about other problems a computer is apt to encounter when graphing functions, and possible solutions.

9. Enter the function F(X)=X^2, set the domain to [0,2], the range to [0,4] and and the screen to CLEAR. Plot the function. Next change the function to F(X)=2*X-1 and the screen to KEEP, then graph again. What you see is the graph of a parabola and its tangent line at the point (1,1). To 'zoom in' on the picture repeat the exercise but use the domain [.9,1.1] and the range [.8,1.2]. The graphs of the parabola and the tangent line will appear almost identical on the smaller interval.

10. (Calculus teachers only.) Graph a function and its derivative on the same screen, then relate the positive and negative sections of the derivative to the increasing and decreasing sections of the original function. For example, graph F(X)=X^2 - 2*X + 2 on the domain [0,2] and range [-2,2] with screen CLEAR, then graph F(X) = 2*X - 2 with the screen set to KEEP. Observe the relationship between the zeros of the second function and the extrema of the first.

Listing of GRAPHIT.

```
10  ONERR  GOTO 20000
20  GOSUB 10000: GOTO 1000
899  REM Compute constants for scaling functions
900 MU = 279 / (B - A):MV = 191 / (C - D):BU =  - MU * A:
    BV =  - MV * D: RETURN
999  REM Main menu
1000  HOME : PRINT  TAB( 15)"AUTOGRAPH": VTAB 3:
      PRINT  TAB( 12)"BY MARK HARRIS": VTAB 6:
      PRINT "1. DOMAIN: [";A;",";B"]": PRINT : PRINT "2. RANGE:   ";
1010  IF AUT THEN  PRINT "AUTOMATIC": GOTO 1030
1020  PRINT "[";C;",";D"]"
1030  PRINT : PRINT "3. F(X)=";F$
1040  PRINT : PRINT "4. ";SC$(CF);" SCREEN"
1050  PRINT : PRINT "5. SHOW GRAPH"
1052  PRINT : PRINT "6. PLOT"
1055  PRINT : PRINT "7. QUIT"
1060  PRINT : PRINT : PRINT  TAB( 10)"WHICH? (1-7) ";
1070  GET A$: ON  VAL (A$) GOTO 2000,2500,3000,3200,3800,3300,3900
      : GOTO 1070
1999  REM Get domain
2000  HOME : VTAB 10: PRINT "LEFT ENDPOINT: ";A;: HTAB 16:
      INPUT "";A$: IF A$ = "" THEN  VTAB 10: HTAB 16: PRINT A:
      GOTO 2010
2005 A =  VAL (A$)
2010  VTAB 13: PRINT "RIGHT ENDPOINT: ";B;: HTAB 17: INPUT "";A$:
      IF A$ = "" THEN  VTAB 13: HTAB 17: PRINT B: GOTO 2020
2015 B =  VAL (A$)
2020  GOTO 1000
2499  REM Set vertical scaling
2500  HOME : VTAB 10: PRINT  TAB( 10)"SCALING = ";MD$(AUT): PRINT
      : PRINT  TAB( 12)"OK? (Y,N) ";
2501  GET A$: IF A$ = "N" THEN AUT = 1 - AUT: GOTO 2500
2502  ON AUT = 1 GOTO 1000
2503  HOME : VTAB 10: PRINT "MINIMUM VALUE: ";C;: HTAB 16:
      INPUT "";A$: IF A$ = "" THEN  VTAB 10: HTAB 16: PRINT C:
      GOTO 2510
2505 C =  VAL (A$)
2510  VTAB 13: PRINT "MAXIMUM VALUE: ";D;: HTAB 16: INPUT "";A$:
      IF A$ = "" THEN  VTAB 13: HTAB 16: PRINT D: GOTO 2520
2515 D =  VAL (A$)
2520  GOTO 1000
2999  REM Enter function
3000  HOME : VTAB 10: PRINT "F(X)=";:U =  PEEK (36) + 1:
      V =  PEEK (37) + 1: PRINT F$;
```

```
3010  HTAB U: VTAB V: INPUT "";A$: IF A$ = "" THEN 1000
3020  CALL 768,A$,10050: REM Put function definition in line 10050
3030 X =  FN F(1.23): REM Test function at arbitrary point
3040 F$ = A$: GOTO 1000
3199  REM Toggle clear/keep screen
3200 CF = 1 - CF: GOTO 1000
3299  REM Plot
3300 H = (B - A) / NN: ON AUT = 0 GOTO 3400:C = 10E37:D =  - C:
     X = A: REM IF AUTOMATIC SCALING, FIND EXTREMA
3310  FOR I = 0 TO NN
3320 F(I) =  FN F(X):D%(I) = 1
3330  IF F(I) > D THEN D = F(I)
3340  IF F(I) < C THEN C = F(I)
3350 X = X + H: NEXT
3400  GOSUB 900: IF CF = 0 THEN  GOSUB 4000: GOTO 3500
3410  HGR2 : HCOLOR= 3:UU =  FN U(0): IF UU >  = 0 AND UU <  = 279
      THEN  HPLOT UU,0 TO UU,191
3420 VV =  FN V(0): IF VV >  = 0 AND VV <  = 191 THEN
     HPLOT 0,VV TO 279,VV
3500 PF = 0:X = A: FOR I = 0 TO NN
3510  ON AUT GOTO 3550:D%(I) = 0:F(I) =  FN F(X):
      IF F(I) <  = D AND F(I) >  = C THEN D%(I) = 1
3550  ON I <  > 0 GOTO 3560: IF D%(0) THEN
      HPLOT  FN U(X), FN V(F(I))
3555  GOTO 3600
3560  IF D%(I - 1) AND D%(I) THEN  HPLOT  TO  FN U(X), FN V(F(I)):
      GOTO 3600
3570  IF D%(I) THEN  HPLOT  FN U(X), FN V(F(I))
3600 X = X + H: NEXT : POKE ST,0: GET A$: TEXT : GOTO 1000
3800  GOSUB 4000: GET A$: TEXT : GOTO 1000
3900  HOME : VTAB 10: PRINT  TAB( 10)"QUIT? (Y,N) ";
3910  GET A$: IF A$ <  > "Y" THEN 1000
3920  POKE 216,0: HOME : END
3999  REM Show high resolution screen 2 but don't clear it.
4000  POKE  - 16304,0: POKE  - 16299,0: POKE  - 16297,0:
      POKE  - 16302,0: RETURN
10000  HOME : VTAB 10: PRINT  TAB( 10)"PLEASE STAND BY...";:
       FOR I = 0 TO 9: READ X: POKE 950 + I,X: NEXT :
       DATA  104,168,104,166,223,154,72,152,72,96 :
       REM Error-fixing routine from Applesoft manual.
10005  DIM SC$(1),F(40),MD$(1),D%(40)
10010 SC$(0) = "KEEP ":SC$(1) = "CLEAR"
10015 MD$(0) = "MANUAL":MD$(1) = "AUTOMATIC"
10020 CF = 1:NN = 40:ST = 12 * 16 ^ 3 + 16
10030 F$ = "SIN(X)"
10040 A = 0:B = 6.2832:C =  - 1:D = 1
10050  DEF  FN F(X) =  SIN (X): REM *******************************
       ******************************
10055  REM *** Important note ***
       Line 10050 is changed by the program during execution.  It
       is important to leave the REM statement long enough (e.g.
       with '*' characters) to allow for the function entered by
       the user.
10060  DEF  FN U(X) = MU * X + BU: DEF  FN V(Y) = MV * Y + BV
```

```
11000  FOR I = 1 TO 141: READ X: POKE 768 + I - 1,X: NEXT : REM
       This is the routine which 'pokes' the function definition
       into the running program.
11010  DATA  32,190,222,32,227,223,132,255,133,254,160,0,177,254,
       72,200,177,254,72,200
11020  DATA  177,254,133,255,104,133,254,104,168,169,0,153,0,2,
       136,177,254,201,33,208
11030  DATA  2,169,0,153,0,2,192,0,208,240,32,190,222,32,12,218,
       165,184,72,165
11040  DATA  185,72,32,26,214,144,69,160,4,177,155,201,184,208,
       61,200,192,20,240,56
11050  DATA  177,155,201,208,208,245,200,152,24,101,155,133,155,
       144,2,230,156,162,255,160
11060  DATA  4,32,108,213,160,0,185,0,2,240,9,145,155,200,192,
       240,240,18,208,242
11070  DATA  169,58,145,155,200,169,178,145,155,104,133,185,104,
       133,184,96,162,16,76,18
11080  DATA  212,4,248,36,216,56,72,173,0,224,73,76,240,3,169,128,
       44,169,130,141
11100  CALL 768,F$,10050
11200  RETURN
19999  REM Error handling routine
20000  CALL 950:R1 =  PEEK (222):
       R2 =  PEEK (218) +  PEEK (219) * 256:
       REM  Error code and line #
20010  IF R2 = 3030 AND R1 = 16 THEN  PRINT  CHR$ (7);: GOTO 3000
20015  IF R2 = 3030 THEN 3040
20020  IF R2 = 3320 THEN D%(I) = 0: GOTO 3350
20030  IF R2 = 3510 THEN 3600
20099  HOME : VTAB 10: PRINT "ERROR "R1" OCCURRED IN LINE "R2".":
       POKE 216,0
```