DOCUMENT RESUME

ED 273 683 TM 860 550

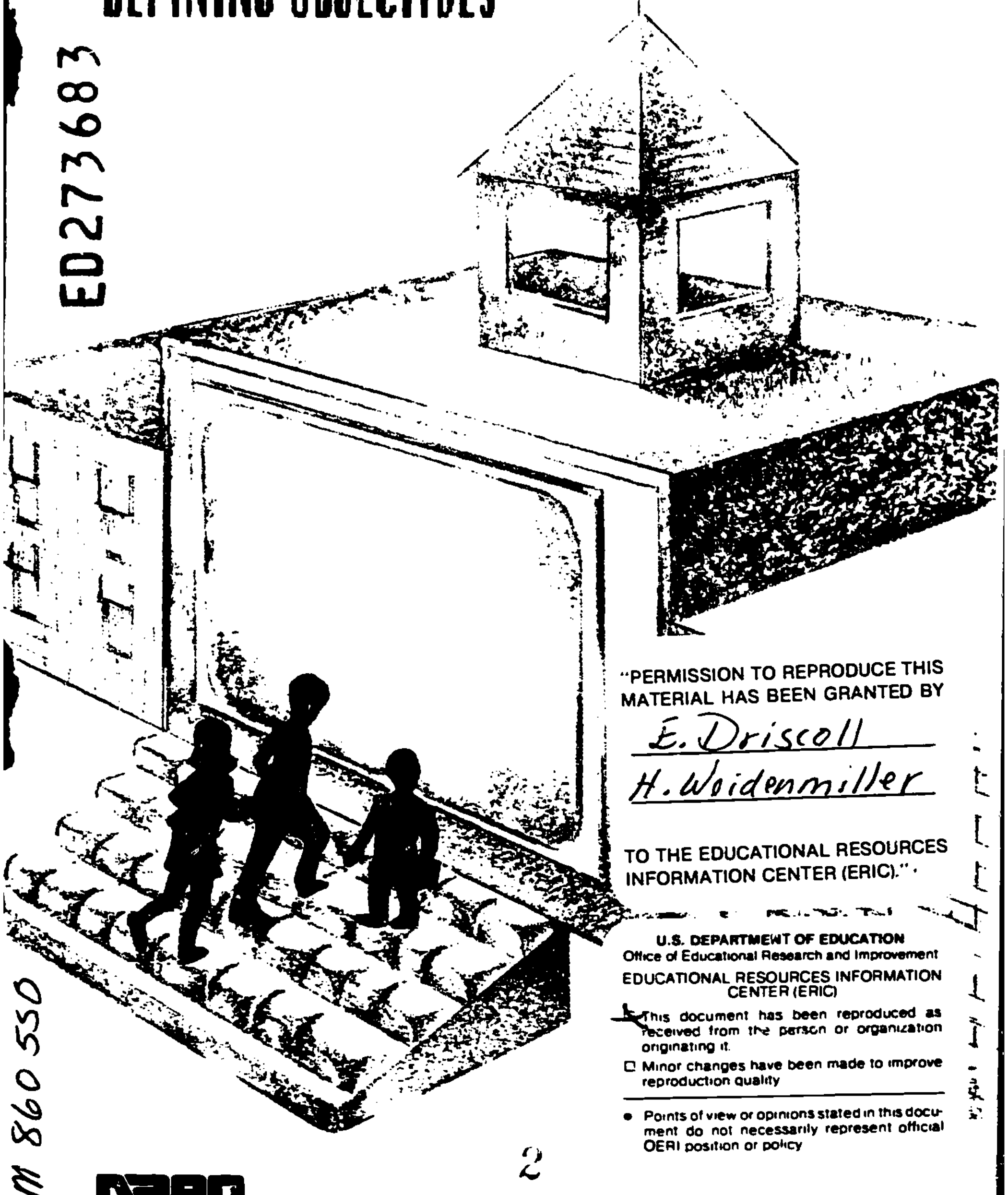
| | |
|---|---|
| TITLE | A Framework for Assessing Computer Competence: Defining Objectives. |
| INSTITUTION | National Assessment of Educational Progress, Princeton, NJ. |
| SPONS AGENCY | Office of Educational Research and Improvement (ED), Washington, DC. |
| REPORT NO | ISBN-0-88685-043-6 |
| PUB DATE | [86] |
| GRANT | NIE-G-83-0011 |
| NOTE | 43p. |
| AVAILABLE FROM | National Assessment of Educational Progress, Educational Testing Service, Rosedale Road, Princeton, NJ 08541 ($5.00). |
| PUB TYPE | Reports - Research/Technical (143) |
| EDRS PRICE | MF01/PC02 Plus Postage. |
| DESCRIPTORS | Academic Achievement; Affective Objectives; *Cognitive Objectives; *Computer Literacy; Computer Science; Curriculum Development; *Educational Assessment; *Educational Objectives; Elementary Secondary Education; Grade 3; Grade 7; Grade 11; *Measurement Objectives; *National Surveys; Problem Solving; Programing; Student Attitudes; Test Construction; Testing Programs; Test Items |
| IDENTIFIERS | *National Assessment of Educational Progress |


ABSTRACT

Computer skills objectives have been developed for the 1986 National Assessment of Educational Progress (NAEP). These items will be administered to a large number of American students aged 9, 13, and 17 in grades 3, 7, and 11. For this first national assessment of computer skills, it was necessary to consider the existing expertise of school staff, the current curriculum, the rapidly changing nature of computer science, and educators' different definitions of computer competence. Generally, educators agree that students need an exposure to computing that enables them both to experience the power of computing and to use that power to solve significant and interesting problems. Measurement objectives involve both paper-and-pencil tests of cognitive ability, as well as the practical ability to use a computer to solve problems. Three categories of cognitive objectives include knowledge, operation, and problem solving and design. Eight applications areas include word processing, database management, laboratory instrumentation, telecommunications, graphics, music generation, spreadsheets, and models and simulations. Programming objectives involve elements of a language; structures of data and control; and program planning, design, and testing. A number of items are illustrated. Five items are also included to illustrate attitudinal objectives. (GDC)

# A FRAMEWORK FOR ASSESSING COMPUTER COMPETENCE: DEFINING OBJECTIVES



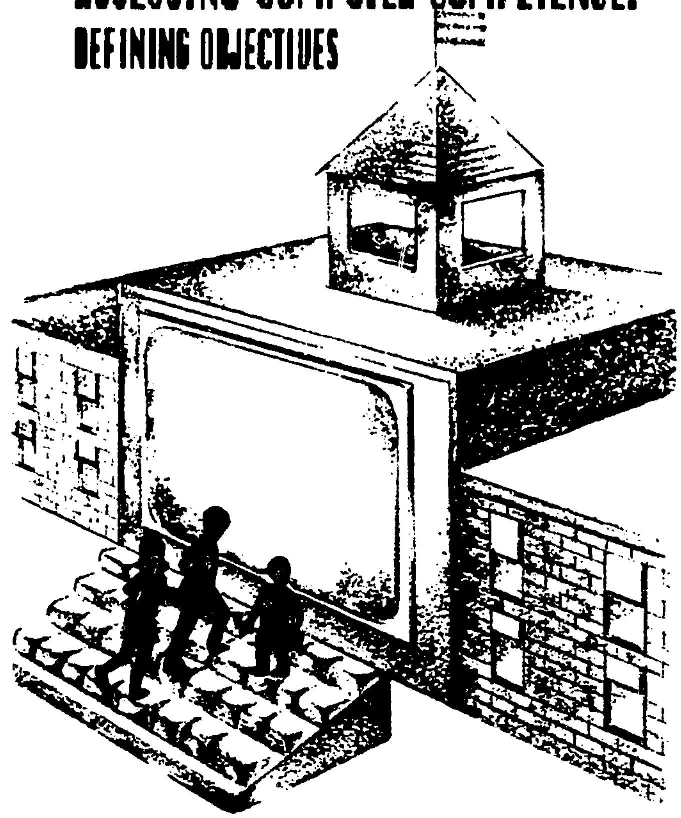






naep THE NATION'S REPORT CARD • EDUCATIONAL TESTING SERVICE

# A FRAMEWORK FOR ASSESSING COMPUTER COMPETENCE: DEFINING OBJECTIVES



National Assessment of Educational Progress
CN 6710
Princeton, NJ 08541-6710

Objectives Booklet No. 17-CC-10

Educational Testing Service

The National Assessment of Educational Progress is funded by the U.S. Department of Education under a grant to Educational Testing Service. National Assessment is an education research project mandated by Congress to collect and report data over time on the performance of young Americans in various learning areas. It makes available information on assessment procedures to state and local education agencies.

This booklet, No. 17-CC-10, can be ordered from the National Assessment of Educational Progress at Educational Testing Service, Rosedale Road, Princeton, New Jersey 08541.

Library of Congress Catalog Card Number 72-169008

Although objectives booklets produced by the National Assessment of Educational Progress between 1969 and 1972 have their own individual catalog card numbers, the number above is a series number assigned to all National Assessment objectives booklets published since then.

ISBN 0-88685-043-6

The work upon which this publication is based was performed pursuant to Grant No. NIE-G-83-0011 of the Office for Educational Research and Improvement. It does not, however, necessarily reflect the views of that agency.

# Table of Contents

# ABOUT NAEP

For almost two decades, NAEP has served as the nation's report card, regularly collecting and reporting information about the knowledge, skills, and attitudes of nine-, 13-, and 17-year-olds in reading, writing, mathematics, science, literature, art, music, social studies, computer competence, citizenship, as well as the career and occupational development of young adults. The NAEP data base is composed of well over a million students.

To measure the quality of education in the United States, NAEP has, through the years, developed educational objectives in all the areas it assesses, and has administered assessments based on these objectives. The objectives are the result of a consensus process incorporating the opinions of concerned educators and citizens representing various backgrounds and points of view.

The questions or exercises written to fit NAEP's objectives are developed by educators from all parts of the country working with specialists at Educational Testing Service. The exercises are evaluated on the basis of their academic appropriateness, their effectiveness, their freedom from bias and stereotyping, and their sensitivity to racial, ethnic, religious, and political groups.

The exercises span a wide range of difficulty and are presented in a variety of formats. Many exercises are of the familiar multiple-choice type; others require the student to write essays or short responses.

The National Assessment reports information about students' learning every other year and provides information by grade (3, 7, and 11) as well as by age (9, 13, and 17).

NAEP is funded by the Office for Educational Research and Improvement and is administered by Educational Testing Service (ETS) as an activity of its Center for the Assessment of Educational Progress.

## The Objectives Development Process

The 1986 computer competence objectives are merely a guide for designing the first national assessment of computer competence. They were developed throughout 1984 and early 1985 by the Learning Area Committee on Computer Competence and the NAEP staff, drawing on the views of many people across the nation. The development process relied upon subject area specialists to ensure that the objectives reflect both current practice and expert opinion. NAEP involved teachers and other school staff with responsibilities at the elementary, junior, and senior high school level to ensure that the objectives are appropriate and realistic from the perspective of the classroom teacher. In addition, NAEP included parents, representatives of the business community, and other concerned citizens in the process to ensure that the objectives are, insofar as possible, free from bias and in keeping with these constituencies' expectations of educational achievement. School superintendents and curriculum specialists also participated in the review process, as did specialists in computer education and persons responsible for educational legislation and policy at all levels of government. Contributors and reviewers were chosen to represent both sexes, a variety of racial and ethnic groups, and communities varying in size, type and geographical location.

## Computer Competence: A New Educational Goal

Through this broad process of involvement, NAEP has tried to reflect accurately the national expectations for students' ability to use computers. It would be foolish to pretend, however, that there is unanimity within and throughout all groups consulted on the question of what students should know and be able to do. Even if there were such agreement, the experience we have had thus far suggests that it would be transitory.

In the context of NAEP's history, the task of producing objectives for an assessment of computer competence has presented an unusual challenge. As recently as 1980, very few people thought schools should teach students the skills necessary for becoming sophisticated computer users. By 1983, however, the National Commission on Excellence in Education, in *A Nation at Risk*, a report to the nation, declared that the ability to use the computer is now a fourth basic skill, on a par with the time-honored three R's.

This sudden rise to prominence of computing in the classroom has caused unique problems for educators. Virtually the same introductory computing skills are being taught today to both young children and corporate executives. Some students know more about computing than do their classroom teachers. Under these circumstances, it is unreasonable to expect a standardized curriculum in which simple skills are taught in the early grades, followed by a smooth progression of increasingly complex material and abilities. Yet, as young children acquire competence, follow-up activities are increasingly needed to build on that competence, thus transforming the curriculum over time. This assessment finds the schools in such a transition. Consequently, it is necessary to assess beginning competence at all grade levels and, at the same time, to measure higher levels of competence at some grade levels.

The world of computing is not likely to stand still while schools develop an orderly curriculum. Between publication of this document and release of the results of the 1986 assessment, changes will take place in the design, cost, and performance of the machines and software that schools can buy. This will have a profound effect on the uses to which computers are put and on the specific skills required to use them.

The rate of change in the world of computing is one factor affecting the choice of objectives for this assessment. Since what schools teach is, in part, a function of the amount of experience they have with computing, a nationwide assessment must attempt to capture the objectives of both the schools just starting out and those that have had more experience. In addition, because this is the first of repeated assessments of computer competence, it must incorporate aspects of computer competence that will be of interest to this country in the future, providing a baseline from which to measure students' progress over time. For both these reasons, this assessment will measure students' abilities in areas that are not currently in the curriculum of many schools and districts. This approach is consistent with NAEP's purpose—to find out what students know and to track changes in their knowledge over time, rather than to measure districts, schools, or students against some norm.

This booklet is designed to do more than just serve as a set of guidelines for developing items for the 1986 assessment. It will also add to the resource material from which schools, districts, and states develop their own programs for improving students' computer competence.

The sections that follow include the full set of objectives resulting from the consensus process. Some of the skills outlined can be measured more easily at the local level than on the national scale, and others are difficult to measure at all. Much of what we think is important to teach in our schools does not lend itself to easy measurement, but we do not forego teaching it for that reason.

# OVERALL PLAN FOR THE ASSESSMENT OF COMPUTER COMPETENCE

## Defining Computer Competence

The computer is a tool of extraordinary power, enabling those who know how to use it to extend greatly their capacity to acquire, analyze, synthesize, and interpret information and to express themselves in countless domains. Computer competence can take many forms, all of which require using the power of the computer to perform tasks like writing, keeping track of a complex inventory of goods, modeling an economy, or analyzing voting behavior in presidential elections.

In practical terms, people acquire mastery of the computer through proficiency in the use of general-purpose programs written by others (such as word processing and database programs), through the writing of programs, and through the modification of their own and others' programs. These activities also require sufficient experience in selecting the hardware and software appropriate for each task and in computer operation.

A general notion of what computing is provides a good start but does not tell us much about what teachers should teach, students should learn, and we should assess. What does it mean to be "computer competent"?

The great majority of educators agree that, for most students, computer competence requires an exposure to computing that enables them both to experience the power of computing and to use that power to solve significant and interesting problems. Some educators, however, would approach school computing as the first step in a progression that, for certain students, ends in the graduate study of computing. Others would place less emphasis on explicit mastery of the formal theories and constructs that underlie computer science and more on the ability to accomplish useful tasks with a computer. These divergent views have nothing to do with differences over whether students' computer studies should be "hard" or "easy", but reflect different purposes and styles of instruction. Whichever view one takes on this question, the study of computing requires considerable rigor and intellectual discipline.

This objectives booklet does not take sides in the debates about school computing. It accommodates different views in a consensus statement about what is the minimum many think should be considered when constructing a curriculum. The committee believes it is important that NAEP be in a position to report on students' computing skills in a way that embraces both the view that students should be learning the rudiments of computer science and the view that students should be able to use computers for a variety of purposes.

In the following pages, we will use the term "computer science" to refer mainly to that part of the objectives statement that is concerned primarily with expressing ideas and solving problems by programming the computer. The term is used mainly to remind the reader that there is more to programming than coding skills and that the intellectual substance of computing extends beyond the narrow definition of programming. These statements hold true no matter how the curriculum of any particular school may be structured.

Though no one expects elementary and secondary school students to study the full range of subjects addressed by college and graduate students of computer science, there is a wide expectation that virtually all students will receive some instruction in the rudiments. It is also true, however, that a steadily growing number of states and districts have goals for student use of computers that call for development of skills often learned and typically used outside of computer science/programming classes. One such skill is students' ability to use applications software—word processing programs, database management programs, spreadsheets, and communications programs, for example—both to accomplish significant tasks in many of their non-computing classes and to prepare them to use these powerful tools after graduation. To cite the most obvious example, some schools are now restructuring the way writing is taught in order to take advantage of the capability of word processing programs to improve students' writing ability.

Knowing something about the structure of computers is also important to users engaged in programming or applications. Objectives relating to the understanding of computer systems appear in all three major subdivisions of the objectives statement.

Finally, the computer is not only a tool to be mastered, it is a powerful influence on people's lives, whether they use the equipment or not. While the primary concern of this assessment is to determine the ability of students to use the computer, it will also attend to students' attitudes toward computing, because these attitudes will have an important influence on their motivation to take advantage of the technology. In addition, the assessment will examine students' knowledge about, and ability to deal with, the issues posed for society by the virtually universal use of computer and telecommunications technology.

These distinctions between computer science/programming, use of applications packages, and knowledge and attitudes about computing are somewhat artificial. Many higher-level programming languages are evolving into software that appears to the user very much like many applications packages, while many applications packages are incorporating features that give them characteristics most people associate with programming languages. The point is that arguments over whether students should be taught the use of applications programs or taught to program appear increasingly to be more a matter of difference in degree than difference in kind, and there is growing evidence that, to the extent they are different, both sets of skills will be very important.

As confusing as the future of computing appears, it is reasonable to forecast that most people using computers will probably run integrated applications packages, but their ability to take advantage of the full power of those programs may well require them to "program" in the language of those applications packages, drawing on the kinds of skills typically taught as part of the computer sciences.

Some readers may wonder whether the computer competence assessment will deal with a common school use of computers not so far mentioned—computer-assisted-instruction. Since the purpose of the assessment is not to assess the value to schools of computers as instructional delivery systems, but rather the competence students exhibit as users of computers, the assessment will not report on the merits of CAI as such. However, to the extent that use of computers by students for CAI improves their competence at using the computer, that learning will be reflected in students' performance on the assessment.

## Measuring Computer Competence

Here we turn to some of the issues arising from what we can assess, how we can assess it, and what we will be able to conclude from the methods available to us. These measurement and interpretation issues significantly affect the shape of the whole assessment.

It is a curious result of our interest in competence that we will often want to know whether students have mastered skills that have very little intellectual content—"trivial" skills, if you will—because mastery of those skills opens up possibilities for using the computer in acquiring more crucial skills.

Many people believe, for example, that students who frequently use word processing software in school may, with an appropriate writing curriculum, become better writers than they would without word processors. Students who become adept users of lab instrumentation software may learn more about laboratory science than those who do not. Students who master database management software may find it much easier to gather and analyze data relevant to courses ranging from economics to construction management than students who do not. In addition, competent use of software in one applications area may require more expertise than in another.

The assessment will include features that, over time, may shed some light on the relationship between computer-related skills and those acquired in other subject areas, and on computer availability and its effect on student performance in school. Students will be asked questions, for example, about the amount of computer time they have at school and elsewhere and about the kind of software they have available to them, and how often and under what conditions they use it. The design of the entire assessment permits correlation of the answers to such background questions with performance on cognitive items in the computer competence assessment and with performance on exercises from other subject areas. The lessons about these relationships may, however, be more indicative than conclusive on the points of interest.

For example, we may find out in a future assessment that those children who have a lot of access at home to computers equipped with word processing software are more competent users of that software than children who do not. We may also find that children who have a lot of access to word processing software at home do better on measures of writing ability than children who do not. We cannot conclude from such data, however, that greater access to word processing software leads to improved writing ability, because it is probable that the affluence implied by computers in the home leads to other advantages enjoyed by these children that contribute to greater writing ability.

These limitations do not render such correlations useless, however. They may make otherwise inexplicable results of the assessment interpretable and, particularly when they produce unexpected results, may lead researchers to lines of investigation that will ultimately produce more informed policy-making. Thus the assessment of computer competence will, in part, be designed to include questions and exercises that will permit connections to be made between such things as resources available to students for computing and the competence in computing displayed by those students.

We turn now to a central issue—perhaps the central issue—for this assessment: how to determine whether someone is a competent user of computers. It is common for teachers of computing to relate stories of students who fail miserably on written examinations in computing but who work well on a computer. At the same time, teachers often speak of students who have a very hard time working on a computer but who do quite well on pencil-and-paper tests of computing ability. It is not hard to construct written questions on whether a student can recall the names of computer parts, specific commands, or pioneers in computing, but that sort of knowledge does not confer competence on a student. Competence means the student can solve real problems with a computer. Defining competence that way, can we adequately assess students' competence without having them demonstrate that competence on the computer?

Although not all, and not necessarily the most important aspects of computer competence can be adequately assessed with paper-and-pencil exercises, the Learning Area Committee on Computer Competence agreed that many aspects of computer competence can be. For example, one can hardly be judged competent as a user of today's computers without knowing how to set up and operate computer systems. This involves everything from turning the system unit on and off to getting the printer to print properly, from properly setting the parameters on communications equipment to copying selected files from one disk to another. The variety of computers, operating systems, and peripheral equipment in use, and the differences among them, are simply too great to permit the writing of questions and scoring of responses that could determine whether students know how to set up and operate a system. In these situations, it is preferable to use performance testing to get at the answers to such questions, using whatever equipment is normally available to the student. Some observers believe that such performance testing is especially important for young children, because measures of computer competence might otherwise be easily confounded with measures of reading ability.

In this performance testing environment the question would be, Did it work? Did the printer print? Did the program run and produce the intended output? Did communication between two linked machines take place? Though there may be great variety in the detailed means for achieving the result, the question is whether the student can master those means to achieve the desired result. Therein lies competence.

There are two issues here. First, given an exercise in which a paper-and-pencil test might plausibly be a valid substitute for a performance test, is it in fact a valid substitute? Second, in some cases no plausible paper-and-pencil test for a given competence can be imagined, let alone validated.

The Learning Area Committee on Computer Competence believes it is important to fully validate all questionable items by requiring students to use a computer to complete each of the items. The Committee also believes that much of what the public reasonably wants to know about students' competence can probably be determined by having the students demonstrate what they can do on computers. This implies the importance not only of having a small sample of students validate the paper-and-pencil test, but, possibly, of having some part of the assessment itself administered with performance exercises.

Performance testing is very much more expensive than machine scoring of multiple choice paper-and-pencil instruments. At this writing, funds are not available to administer performance exercises in the first year of the computer assessment. It will be very important to take that into consideration when interpreting the assessment results. This is true for all age and grade levels, but particularly at the age 9/grade 3 level, where there is the greatest danger of confusing computing ability with reading ability.

One final note on paper-and-pencil tests: however desirable performance testing might be in the light of the limitations of paper-and-pencil tests, it is not a panacea. Performance contexts may differ widely and observers may make very different judgments on the same performance. Thus issues of validity and reliability do not disappear when performance tests are used. With this caution in mind, the committee encourages local school officials to consider the use of performance tests whenever it is appropriate and feasible to do so, since performance testing becomes steadily more feasible as the unit tested moves from the nation to the individual classroom.

Another issue of measurement, higher-order skills, is closely allied with the issues involved in assessing performance. The committee sought to create an assessment that could report on the degree to which students truly understand the material to which they have been exposed, and on their ability to apply that understanding to real and intellectually challenging tasks. This capacity enables students to analyze unfamiliar tasks and synthesize solutions from various sources of knowledge. These "higher order skills" are widely held by educators and the general public to be crucial underpinning for our graduates' future growth. Higher-order skills are important in every major discipline, including computing.

To test students' competence in using such skills, some items in the assessment will cover familiar ground in an unfamiliar way. Students may be asked to use what they know to design a solution to an unfamiliar problem, to describe a system component in a way that tests their understanding of that component's function, or to analyze some aspect of a policy choice about the rules governing the public use of computers. An assessment that asks students to go well beyond recall of class notes is necessary if we are to investigate student ability either to use these tools or to make wise decisions about how society uses them.

## Three Major Categories of Objectives

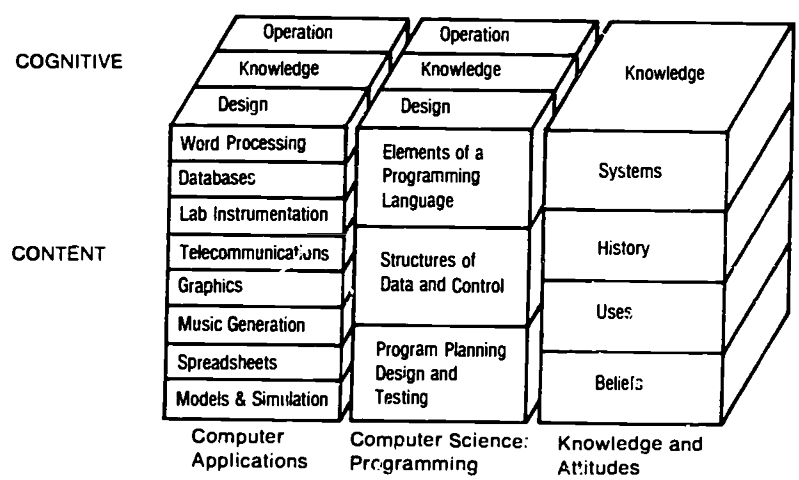
Apart from background questions, the 1986 computer competence assessment will be composed of 309 different exercises to be administered across three age and grade levels of students:

- Age 9/grade 3
- Age 13/grade 7
- Age 17/grade 11

Three major categories of objectives will be assessed at each of these levels. Each of the three categories will further be subdivided into several cognitive levels and content areas (see figure 1).

**Figure 1**

**COMPUTER OBJECTIVE CATEGORIES**




The number of questions in each of the major categories will not be the same at each age/grade level. Programming skills will be most heavily assessed at the age 13/grade 7 level, in keeping with the evolving pedagogic practice of concentrating programming instruction at the junior high school level. Skills in using applications programs will be most heavily assessed at the age 17/grade 11 level, at which students are being introduced to the widest range of applications programs and their uses by most school districts.

Although 540 exercises will be administered across the three age/grade levels, there will be extensive overlap across levels (e.g., the same exercise will be taken by seventh and eleventh graders). This will enable analysts to compare the performance of students at different age/grade levels on identical exercises.

## Cognitive Levels

The exercises in the three categories of the Assessment will be distributed among the following cognitive levels:

- Knowledge
- Operation
- Problem Solving/Design

KNOWLEDGE refers to the recognition or recall of specific ideas, facts and procedures. The emphasis is on a grasp of the meaning and intent of a communication. Remembering is the major cognitive process involved. In the *Computer Applications* category, knowledge includes such things as ability to identify common terms and command names and to distinguish between the functions of different applications packages. In the *Computer Science: Programming* category, it typically includes the ability to recall the syntax rules of a programming language and the names and uses of commands, primitives, and functions. In the *Knowledge and Attitudes* category, it includes the ability to identify computer system components and their uses, to express beliefs about the implications of computer technology and to recall facts about the historical development of computers.

OPERATION refers to routine manipulation of symbols and procedures required to complete specified tasks. Exercises that assess operation will assume that standard techniques have been learned and practiced, and will measure both proficiency in using those techniques and the understanding of how they work. Students assessed at this cognitive level will be asked to apply these techniques to familiar problems. Exercises that assess operation require a sequence of processes involved in the formulation and solution of problems, including recalling and recording knowledge and selecting and carrying out procedures.

PROBLEM SOLVING/DESIGN refers to more complex intellectual processes than those routinely involved in procedural skill application and elaboration of conceptual understanding. Problem-solving exercises would require such behaviors as identifying and using a problem solving strategy, differentiating relevant from irrelevant information, formulating a problem or modeling a problem situation, determining what information is necessary to solve a problem, or organizing given information to represent a problem. These exercises would also include such behaviors as drawing generalizations or testing their validity, recognizing patterns and describing or symbolizing their relationships, conjecturing, informally making inferences or drawing conclusions using inductive or deductive processes, analyzing relationships among data, or identifying and evaluating alternatives.

# OBJECTIVES FOR COMPUTER APPLICATIONS

## Introduction

Because programming will be covered in the computer science portion of the assessment, exercises in the applications areas will not involve the writing of programs but will, instead, assess students' ability to use programs that other people have written. These exercises will be carefully written and reviewed to ensure that they do not assume familiarity with any particular commercial package.

Items administered in the *Computer Applications* section of the assessment will be designed on the assumption that students are using these computer applications as a part of the typical school curriculum. For example, an elementary school student might be asked how a word processor could be used to write and edit a story normally written without the aid of a computer; a junior high school student might be asked how a database management package could be used to manipulate social studies data; and a high school student might be asked to show how laboratory instrumentation software could be used to take measurements and display the results in an experiment from a typical high school biology course. The substantive, non-computing content of these items will not be demanding, so that, insofar as possible, what will be assessed will be computing competence rather than competence in some other subject. The items will, however, be framed from the viewpoint that applications programs are only tools for use in other subject areas. Competence in using these tools will mainly be assessed by creating tasks for the student to perform that parallel real tasks they are likely to encounter in their scholastic and extracurricular activities.

## Content Areas

The eight applications areas that have been selected for inclusion in this objectives booklet are:

- word processing
- database management
- lab instrumentation
- telecommunications
- graphics
- music generation
- spreadsheets
- models and simulations

They were selected on the basis of their reported use in some schools. No assumption has been made that all eight are in widespread use at the publication time of this objectives booklet. Background questions will be asked in the assessment that will enable NAEP to report on each area's frequency of use at the time the assessment is administered.

WORD PROCESSING software includes systems for creating and manipulating text and is generally not viewed as an end in itself but as a means through which teachers can help students improve their ability to express ideas clearly, concisely, and forcefully in written form. Students typically use word processing to create, store, retrieve, edit, revise, format, and print text in creative writing, report preparation, school journalism, and other written assignments.

Students who have mastered word processing are expected to know what the capabilities of such software are and what features are appropriate for different tasks. For example, in a text document, students may need to include a section of a spreadsheet from another file, incorporate an illustration created with a graphics program, or insert names and addresses for form letters from a database. They may need to use various kinds of associated utility programs, such as outlining tools, spelling checkers, electronic thesauruses, punctuation checkers, and writing style programs.[1]

[1]Following each subsection of the objectives, illustrative exercises are provided. These exercises are offered to give a concrete example of a task or groups of tasks a competent student would be able to accomplish. No one exercise can be designed to illustrate the full range of tasks implied by the preceding narrative, nor does any example encompass all levels of difficulty associated with any area of performance. Some of the illustrative exercises are similar to those that will be included in the assessment itself; others are in the form of performance tasks that could be assigned by classroom teachers. In the latter case, performance of the indicated tasks may take considerably more time than would be available for an exercise in the national assessment. Actual exercises in the national assessment will be designed to assess much the same skills, but will use different techniques to do so.

ILLUSTRATIVE EXERCISE #1
for grades 3 and 7

The following paragraph is on your screen:

> The dog is running toward the house. Rover stops
> when he heers his name calld.
> Jack calls out. "Rover"

A. Correct the spelling.
B. Take the last sentence and put it after the first sentence.

DATABASE MANAGEMENT software makes possible a systematic approach to organizing, storing, updating and retrieving information. Use of a database often makes it easier to find, display, and interpret data in cases where a large amount of data is involved. Generating a useful database requires attention to the structure of the data in relation to the uses to which it is likely to be put.

Typically, student activities involving database management might include construction of a computer-based bibliography for a research project, creation of a database cataloging a record collection, or design of a database for storing ticket sales information for a regular school event.

A knowledgeable user of databases is acquainted with the general functions, advantages, and characteristics of database management programs and with common database management terminology. Such knowledge, however, does not make one a competent user of database programs. The competent user is one who can design and create a database to solve a real problem or perform a real function. This involves defining the problem to be solved, identifying and classifying the data relevant to that problem, and designing the form in which the data is to be entered and reported. To accomplish the task efficiently, the designer has to be able to recognize the tradeoffs involved in the design task by anticipating the uses to which the data will be put and by taking maximum advantage of the specific characteristics of the database management program being used.

Students may also have to know how to protect files from unauthorized use, to select the form of storage to be used, and to design databases to conform to the limitations of both the active storage and mass storage available. It may be important for them to be able to combine information generated by their database program with information generated by other applications programs, such as word processors or spreadsheets. If the best way to display the data they have collected is in graphics form, students will have to know how to transfer the relevant data to the graphics software. In addition, students should have enough familiarity with their data to be able to make estimates of expected results.

ILLUSTRATIVE EXERCISE #2
for grade 11

On the attached sheet is a listing of the countries in South America. Beneath the name of each country is a list of important facts about that country, including its population, gross national product, and principal exports. Your task is to use the computer, software, and printer at your workstation to create a database for this data and then, using that database, to print a report that lists all those countries that export coffee, arranged in order of increasing population.

LABORATORY INSTRUMENTATION software enables students to use a computer and specialized input-output devices as laboratory tools to control experiments and to record and analyze data produced by those experiments. Typical kinds of measurements for which students use lab instrumentation include measurements of temperature, light intensity, sound intensity, pitch, force and time. Students familiar with laboratory instrumentation software should be able to describe the capabilities and typical uses of that software.

Competent users of laboratory instrumentation software are able to correctly select and connect the analogue and digital devices required for instrumentation purposes, to calibrate instruments, to use the equipment and software to control laboratory devices, to monitor and collect instrument output, and to display, store, and analyze data. Depending on the actual uses of laboratory instrumentation software, the competent student would be able to set up and analyze the results of computer-controlled experiments using some combination of light sensors, microphones, temperature sensors, ph sensors, voltage sensors, and moisture sensors.

The following exercise might be appropriate for students in a physics class who are using lab instrumentation software to record and display data produced by heating different materials.

---

### ILLUSTRATIVE EXERCISE #3
for grade 11

---

At your workstation you will find a computer, instrumentation software, a source of heat, a good conductor of heat, insulation material, and a piece of glass. The heat conductor is 2" from the burner. On the other side of the conductor is the insulating material, and the piece of glass is against the insulating material. Connect the heat sensor to the glass and connect its output to the computer. Use this setup to display on the screen of your computer the rise in temperature of the glass in the first minute after you turn the burner from the off position to full on. Let the apparatus cool to room temperature. Then remove the insulating material and repeat the experiment. Record below the difference in the temperature of the glass under the two conditions at ten second intervals from time zero.

| | 10 | 20 | 30 | 40 | 50 | 60 |
|---|---|---|---|---|---|---|
| With Insulation: | | | | | | |
| Without Insulation: | | | | | | |

time in seconds

---

TELECOMMUNICATIONS software and equipment allow students to use computers to send and receive text, graphic images, and other forms of data over telephone lines. Telecommunications typically involves connections between microcomputers, or between a microcomputer and a remote mainframe computer.

Knowledgeable students who will use computers for telecommunications will be able to use the telecommunications software and the peripheral equipment typically required for telecommunications applications. In addition, they will be able to describe the components and functions of computer-based telecommunications systems and to demonstrate familiarity with basic telecommunications terminology.

Competent students will be able to select and connect the components required for particular communications tasks and set the communications parameters appropriately. They will be able to make appropriate tradeoffs between speed and cost when selecting communications equipment and services for particular purposes. In addition, depending on the students' needs and available resources, they may need to access and use effectively some combination of the following telecommunications resources: electronic mail, bibliographic services, news retrieval services, bulletin board systems, and computer conferencing systems.

ILLUSTRATIVE EXERCISE #4
for grade 11

(The student is presented with a shelf full of disconnected components, a computer, a telephone, several software programs, and appropriate documentation.)

Your task is to log on to (here an information utility is named), retrieve the latest story from one of the newswire services on (named subject) and store that story on disk.

The telephone number of the service is ( ). Your ID will be ( ) and the password is ( ).

In order to accomplish that task, you will have to select the appropriate hardware and software, connect the components together correctly, set the communications parameters on the software, go through the log on sequence, find the story, store it on your disk and sign off.

COMPUTER GRAPHICS refers to the use of computers to generate visual displays other than standard text or numbers. Students use computer graphics both to generate graphs and to express themselves in the visual arts.

Graphing is the creation of visual representations of quantitative data and their relationships. Competent students might use graphing software to create graphic displays of economic and population trends, the results of plant growth experiments, or the sales totals by month for advertisements in the school newspaper.

ILLUSTRATIVE EXERCISE #5
for grade 11

The attached sheet shows the costs of producing the school yearbook for each of the last five years and the total amount of money collected from students in each of those years through yearbook sales and advertising. Use the computer and software in front of you to create a bar chart from these figures with the quantities clearly labeled, arranged so that the viewer can visually compare costs and sales easily for each year as well as follow the year to year trends.

Computer-based visual arts involve using the computer to create visual images with appropriate applications software and various input devices such as light pens, drawing tablets, and graphic pads. Students who are competent users of such software and equipment might use it to create sketches, architectural and engineering drawings, illustrations, and animation.

ILLUSTRATIVE EXERCISE #6
for grades 3 and 7

Use the computer and drawing pad in front of you to make a drawing of a house on a field. Show the sun in the sky. When you are done, change the color of the sky and print your drawing.

MUSIC GENERATION software, combined with a computer and appropriate peripheral equipment, enables the student to compose, edit, store, and play computer-generated music.

Knowledgeable students can identify the software and hardware components required for music generation, explain how they work, and describe their functions.

Competent students can assemble music generation systems, connect together the components required for a specific purpose, use that equipment to create a musical score—including scoring for particular instruments and changes in key and tempo—and edit and revise the score to realize a musical idea. They are able to store and retrieve these compositions and to play them through the audio output facilities of the computer system they are using.

The following exercise might be appropriate for music students who have access to music generation software and equipment.

ILLUSTRATIVE EXERCISE #7
for grades 7 and 11

You will find four bars of music printed on the attached sheet. Your task is to use the software and hardware available at your workstation to enter this section of musical score into the memory of your computer, store it on your disk, retrieve it from the disk to the active memory of the computer, change it to the key of C, play the result through the computer, and store the result on your disk.

SPREADSHEETS are high-level languages that can be used to create dynamic models of relationships among quantities. They are typically used in business to test the effects of alternative assumptions on outcomes for the business—for example, the revenue derived from sales of a range of items resulting from different assumptions as to marketing and sales costs for those items.

Knowledgeable students would be able to describe the principles of spreadsheet construction and the use of templates. They would also know about uses to which spreadsheets are typically put and have familiarity with the commands and operations associated with one or more packages of spreadsheet software.

Given a particular problem to solve, competent students would be able to construct a conceptual model of the environment by specifying the relevant variables and their relationships, create a spreadsheet simulation of that model, enter the data, and interpret the results. They would also be able to design their own templates, use packaged templates designed for use with general purpose spreadsheets, and transfer data between spreadsheet programs and other kinds of programs, such as graphics and database packages.

ILLUSTRATIVE EXERCISE #8
for grade 11

The attached sheet shows the budget for the school band for the month of January. Using your computer and spreadsheet software, create a spreadsheet for the budget for the whole year. Assume that there are no additional purchases of uniforms or equipment during the year; uniform cleaning costs stay the same until March, when they will go up by ten percent; transportation costs go up steadily at a rate of .2% each month; and faculty advisor salaries are increased 8% effective September 1. You should show no costs for cleaning, transportation or faculty advisor salaries for July or August. All the categories of cost not mentioned are constant through the year. Your spreadsheet should have cells showing total costs for each month, total costs for the year for each category and total costs for the year. What would the effect on the total budget be if transportation costs declined by .1% each month instead of rising at .2%? If faculty advisor salaries did not increase at all? If there was an unexpected increase in sheet music costs in September of 15%?

MODELS AND SIMULATIONS are categories of software that enable the computer to be used as a tool for simulating real systems, either to understand the behavior of the system or to evaluate strategies for the operation of the system. Electronic spreadsheets fall in this category and can be used to model or simulate many different systems. Typically, however, software for models and simulations is designed to model or simulate a single system, such as an economy, an electric generating plant, or the ecological behavior of an estuary.

Most models and simulations used by elementary and secondary school students require only that the student enter data or commands when prompted to do so by the program, and demand little or no prior knowledge of the modelled system for operating the software. It may be important, however, for the student to know what is involved in modelling or simulating phenomena in general, and to be able to judge the strengths and limitations of particular models based on the degree to which those models capture the relevant variables in the system being modelled.

The following exercise might be appropriate for students familiar with the popular game LEMONADE, a simulation of the economy of a lemonade stand.

ILLUSTRATIVE EXERCISE #9
for grades 7 and 11

I. Which of the following factors that might affect sales of lemonade at the stand is *not* taken into account by the model used to construct the lemonade game?
   A. Whether or not it is raining.
   B. The price of the lemonade.
   C. How attractive the sign advertising the lemonade is.

II. Do you think it is possible to make a lemonade game for the computer that would predict profits and losses for a real lemonade stand with perfect accuracy? Write a brief essay giving your answer and explaining why you think so.

# OBJECTIVES FOR COMPUTER SCIENCE: PROGRAMMING

## Introduction

Experience in writing even very simple programs can be an effective way to learn about computer systems, their principles of operation, their power, and their limitations. A student can read and listen to definitions of terms such as input, memory, processing, and output, but these ideas become far more meaningful after he has written and run a simple program that instructs the computer to request two numbers as input, add them, and display the result.

On a more fundamental level, computer programming provides a framework for expressing ideas about processes and methodology (how to go about performing a task), akin to the way in which mathematics provides a framework for expressing logical truths. Exposure to programming, like exposure to mathematics, provides a perspective and a set of organizational techniques that can be useful in confronting a vast range of complex problems. For these reasons, an assessment of computer competence will include many items that sample the student's ability to understand and to use a programming language.

There is a danger here that "programming" will be understood in the sense of "coding"—translating well-specified problems into one or another computer language on an element-by-element, syntactic basis; and indeed, some items on the assessment will be concerned with the vocabulary and syntax of programming languages. However, the purpose of learning the elements of a programming language is not to just learn the rules of spelling and syntax, but to be able to express ideas and solve problems. Many items on the assessment will deal with programming methodology, general ability to carry out a programming project, and choice of appropriate algorithms, data structures, and control structures. Still other items, some of which are expressed in a programming language, are not about programming at all, but about the organization of computer systems.

In order to stress the need to look beyond "coding" skills, the Learning Area Committee chose to refer to this portion of the assessment as "Computer Science," even though many concerns of computer science—including such topics as hardware architecture or theory of computation—are not addressed in the assessment. As with the applications portion of the assessment, the notion of competence in computer science and programming is expressed in terms of performance.

The objective of this assessment is to determine to what extent students can do programming, rather than to what extent they know about these issues. One should expect that students will demonstrate competence only if they have significant opportunity to design and implement programs.

This portion of the assessment will be given in alternate versions, corresponding to the most popular computer languages currently taught in schools. Students in grades 3 and 7 will have a choice of working in BASIC, Logo, or both, and those in grade 11 in BASIC, Pascal, or both.

## Content Areas

Competence in *Computer Science: Programming* will be assessed in terms of three stages of increasing sophistication in dealing with computer programs:

- Elements of a Programming Language;
- Structures of Data and Control; and
- Program Planning, Design, and Testing.

At each grade level, the assessment will include items pertaining to each of these stages.

ELEMENTS OF A PROGRAMMING LANGUAGE, for the programming language of choice (BASIC, Logo, or Pascal), includes familiarity with elementary statements and commands, as well as the ability to use these to construct and to analyze simple programs that manipulate numeric or alphanumeric data, and (in Logo) to draw pictures using "turtle graphics."

## ILLUSTRATIVE EXERCISE #10

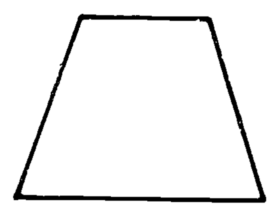
Programming Language Elements — BASIC, for grades 7 and 11

Here is a program that asks for the height and base of a rectangle and prints the area of the rectangle:

```
10 REM PROGRAM AREA
   20 PRINT "WHAT IS THE HEIGHT?"
   30 INPUT H
   40 PRINT "WHAT IS THE BASE?"
   50 INPUT B
   60 PRINT H * B
70 END
```

The area of a trapezoid can be computed in terms of the height, the base, and the top by the formula

$$\text{Area} = \frac{1}{2}\text{Height} \times (\text{Top} + \text{Base})$$



Write a program similar to the one above that asks for the height, base, and top of a trapezoid, and prints the area of the trapezoid.

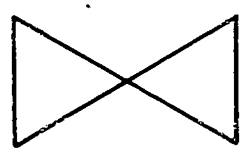
ILLUSTRATIVE EXERCISE #11

Programming Language Elements — Logo, for grades 7 and 11

Here is a procedure that draws a triangle:

```
TO TRIANGLE :SIZE
  REPEAT 3 [FORWARD :SIZE RIGHT 120]
  END
```

Using TRIANGLE, write a procedure called BUTTERFLY, that draws the following picture:



ILLUSTRATIVE EXERCISE #12

Programming Language Elements — Pascal, for grade 11

The following program segment is supposed to swap the values of two integer variables, S and L, if the original value of S is larger than that of L. S and L have been declared previously.

```
IF S < = L THEN
  BEGIN
    S := L;
    L := S
  END
```

This program segment does not work as intended.

(a) If S initially has the value 10 and L initially has the value 5, what will the new values of S and L be after the segment is run?

(b) Modify the segment to work as intended.

STRUCTURES OF DATA AND CONTROL includes using the control structures provided by the language of choice in order to go beyond simple sequential, non-conditional program design, as well as using data structures to manipulate data in aggregates, rather than on an element-by-element basis. Particular topics in this area depend upon the programming language chosen. In BASIC, for example, this category includes structuring programs in terms of decision blocks and iteration blocks and working with arrays; in Logo, facility with common patterns of iteration and recursion, and with list structure; in Pascal, typical control structures, record structures, and data types. In all languages, it includes modularizing programs using subroutines and parameters.

ILLUSTRATIVE EXERCISE #13

Structures of Data and Control — Pascal, for grade 11

(a) Suppose you are given a function called VOWEL, which tests whether a given letter is a vowel (A, E, I, O, or U). Using VOWEL, write a procedure called NVOWEL, that takes a word as input. If the word begins with a vowel, then NVOWEL should output the word with the letter N added onto the beginning. Otherwise, NVOWEL should output the word unchanged. Thus, NVOWEL ("ATE") should output NATE and NVOWEL ("CAT") should output CAT.

(b) The procedure ADDN is supposed to apply NVOWEL to all items in a list of words. For example,

ADDN ("A CAT ATE A MOUSE");

is supposed to output the list

NA CAT NATE NA MOUSE

Write the procedure ADDN.

PROGRAM PLANNING, DESIGN AND TESTING includes demonstrating the ability to carry out a substantive programming project, transforming an initial project description into a specification suitable for implementation as a program, selecting appropriate data structures and algorithms, choosing appropriate program development strategies (e.g. top-down vs. bottom-up), demonstrating effective organizational techniques, and exhibiting testing and debugging skills.

ILLUSTRATIVE EXERCISE #14

Program Planning, Design and Testing — BASIC, for grade 7

A game program has a subroutine at line 1000 for printing instructions. The subroutine starts by doing the following two things:

1. Prints the message "Do you want instructions?"
2. If the player types Y or N, the program continues. If the player types anything else, the program keeps printing "Please enter Y or N" until one of these letters is entered.

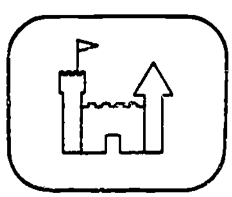
Write this part of the subroutine.

ILLUSTRATIVE EXERCISE #15

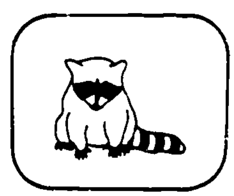
Program Planning, Design & Testing — Logo, for grade 3

Amanda and Jen wanted to write a program to make the turtle draw a picture. They already had procedures for drawing circles, squares, and triangles. The girls drew pictures on paper, showing what new picture they wanted to make.

Amanda wanted to make the turtle draw a castle, like this.



Jen wanted to make the turtle draw a raccoon, like this.



Which picture would it be easier to make the turtle draw? Why?

ILLUSTRATIVE EXERCISE #16

Program Planning, Design & Testing — Pascal, for grade 11

Your school wants you to set up a system on the computer so that a teacher can type in a student's name and see the classes that the student is enrolled in—or the name of a class and find out all the students in that class. The teacher also needs to be able to enter names of students when a course starts and to add names later on.

Here is an example of how the system might work:

```
?STUDENT (S) OR CLASS (C)?
S
NAME? John Smith
GEOMETRY
ENGLISH 24
WORLD HISTORY
CERAMICS

?STUDENT (S) OR CLASS (C)?
C
NAME? GEOMETRY
Herman Brown
Mary Jones
Winifred Miller
John Smith
.
.
.
```

You're going to implement this system in Pascal.

(a) Assume that teachers are almost always going to be printing class lists rather than lists of courses for given students. What would be a good way to represent the data? Write a TYPE statement to describe an appropriate record structure to use.

(b) Describe the major procedures that you would use in your program.

(c) Assume instead that teachers are usually going to be printing lists of courses for given students, rather than class lists for courses. How would this change your answers in parts (A) and (B)?

## ILLUSTRATIVE EXERCISE #17

Program Planning, Design & Testing — BASIC, for grade 11

A computer has a thermometer attached to it as an input device, which will be used in experiments to monitor temperature. In the first experiment, the thermometer will be plunged into a glass of cold water and the temperature readings will be observed.

Here is a program to be used in performing the experiment. It prints the temperature at one-second intervals. The program uses the function T, which reads the thermometer and returns a number giving the indicated temperature, a subroutine at line 100, which prints the value returned by the function T, and a subroutine at line 200, which makes the computer wait for one second.

```
10 REM PROGRAM TEMPERATURE
20   FOR I = 1 TO 10000
30     GOSUB 100
40     GOSUB 200
50   NEXT I
60 END
70
100 REM PRINT THE TEMPERATURE
110   PRINT T
120 RETURN
130
200 REM MAKES THE COMPUTER WAIT ONE SECOND
      .
      .
      .
250 RETURN
```

(a) Rewrite the subroutine at line 100 so that the program will print the time elapsed (in seconds) as well as the temperature.

The result should have the form:

TIME: 1 TEMPERATURE: 75.2
TIME: 2 TEMPERATURE: 74.8
TIME: 3 TEMPERATURE: 74.5

and so on.

(b) Write a program like the one above, but which also prints the change in temperature from one second to the next. The result should have the form:

TIME: 1 TEMPERATURE: 75.2 CHANGE: 0
TIME: 2 TEMPERATURE: 74.8 CHANGE: -0.4
TIME: 3 TEMPERATURE: 74.5 CHANGE: -0.3

and so on.

(c) The experimenter is interested in seeing how long it takes the temperature readings to stabilize. For the purposes of the experiment, the temperature is considered to be stable when the total temperature change over the past 10 seconds is less than 1 degree.

How would you write a program that reads the thermometer at one second intervals until the temperature readings have stabilized, then prints the initial temperature, the final temperature, and the time required for the readings to stabilize?

You don't need to write any BASIC code here, but be sure to describe the major parts of the program and the major routines that must be written. In your answer, you should deal with the following issues:

A. What information does your program keep track of in order to determine if the temperature has stabilized?

B. How does the program keep track of this information? Will you use an array, one or more numeric variables, or another type of data structure?

C. Describe the operations that the program performs in order to keep this information up to date from second to second.

D. Precisely how does your program use this information to test whether the temperature has stabilized? What is the computation performed to make the test?

# PART IV

## OBJECTIVES FOR KNOWLEDGE AND ATTITUDES ABOUT COMPUTERS AND COMPUTING

This statement of objectives has emphasized the importance of the ability to use computers to accomplish significant tasks. There is wide agreement, however, that additional factors must be assessed if the nation is to find out what it wants to know about students, schools, and computing.

First, many people want to know whether students have an understanding of how a computer operates and what the functions of its principal components are, because they feel knowledge of computer systems helps students to perform particular computer operations and to make better judgments about what computers can accomplish.

Second, there is a widespread expectation that schools will play an important role in helping students to understand the ways in which modern information technology is transforming work, education, entertainment and many other everyday activities in our society. Closely linked to this expectation is a strong desire that students understand the nature of the choices that individuals and the society must make as the technology advances, particularly the moral and ethical aspects of those choices.

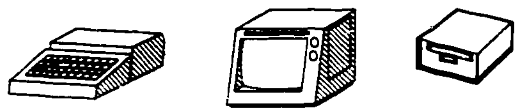
Assessing students' knowledge in this area poses very difficult problems, particularly since there is strong disagreement among knowledgeable people as to what the facts are. This is not a realm in which questions can be asked to which students can reasonably be expected to produce "right answers," but it is important that students understand the nature of the moral and ethical issues involved, that they develop the skills required to analyze carefully the facts presented and arguments made by advocates on all sides of the issues, and that they have the ability to arrive at well reasoned conclusions.

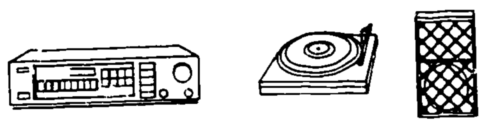
Finally, there is considerable interest in students' attitudes towards computers and computing. This, too, is a difficult area to assess. People are interested in knowing whether students' attitudes towards computers and computing explain in some measure the variation in their competence in computing. Research to date, however, shows that virtually all students think that computers are and will continue to be very important in society and in the workplace. Because there is little or no variance on this attitude measure, it cannot, therefore, be used to explain the variance in any other measure.

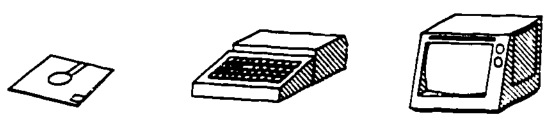
What may be important, though, is variation in students' attitudes toward their experience with computing in school. Students may think that computing is important, but have very little confidence in their ability to accomplish in school what is expected of them. They may think that computing is important to learn, but may believe that what they are learning in school is not what they will have to know about computers later on in school or the workplace. They may believe that they are denied computing opportunities in school that students of another race or sex enjoy. Or they may find the school computing curriculum unchallenging, but look forward with enthusiasm to an opportunity to set their own objectives in the school computer lab after school hours. Knowing what students' attitudes are on these points may well prove useful in determining how best to interpret otherwise baffling variations in performance on the direct measures of computer competence.

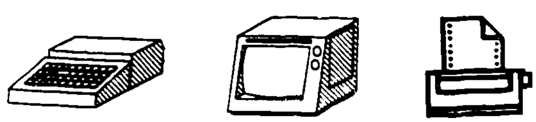
## ILLUSTRATIVE EXERCISE #18
for grades 3, 7 and 11

There are four drawings below. Which one is a drawing of a computer with a monitor and a disk drive?

A. 

B. 

C. 

D. 

ILLUSTRATIVE EXERCISE #19
for grades 7 and 11

Some experts say that the widespread use of computers and robots will lead to increasing unemployment. Others say that the widespread use of computers and robots will not lead to more unemployment.

Since the experts disagree, what is the best way to find out what will happen?

A. Evaluate the methods used by the leading experts, examine their arguments and come to the conclusion that seems most convincing as a result.

B. Consult the encyclopedia under the heading "Computers and Employment".

C. Ask the teacher you think knows the most about computers.

D. None of the above. If the experts do not agree, then it does not make sense for the average person to think about it.

ILLUSTRATIVE EXERCISE #20
for grade 11

Write a two-page essay comparing the development of the computer to the development of the steam engine, the electric motor OR the telephone. You may consult books to gather information for your essay. Your essay should focus on at least one of the following topics:

A. The similarities and differences between the two technologies in the way they affected the kinds of work people do.

B. What the early developers of each machine thought they would be used for, whether they were right and what we can learn from them about our ability to predict the influence of technology on society.

C. What the development of the two technologies teaches us about what it takes to turn a good idea for an invention into a product that influences the lives of many people.

ILLUSTRATIVE EXERCISE #21
for grades 7 and 11

It is illegal to copy much of the software sold for use on micro-computers, but many people copy this software anyway. Do you agree or disagree with the following statements?

A. It is all right for schools to copy this software because they cannot afford to buy it and need it to educate students.
Agree: ( ) Disagree: ( )

B. No one should illegally copy software because that is stealing and stealing is wrong.
Agree: ( ) Disagree: ( )

C. If I bought and paid for software that is not supposed to be copied, I can make a copy for a friend, because I paid for my copy, and my friend would do the same for me.
Agree: ( ) Disagree: ( )

ILLUSTRATIVE EXERCISE #22
for grades 3, 7 and 11

Which of the following statements BEST describes the way you feel about using computers in school?

A. I like computers, but other students seem to get to use them more than I do.

B. I like computers. Learning about them is one of the best things about school.

C. I do not like computers and I wish I did not have to use them at school.

D. I think I would like computers, but I do not get to use them at my school.

E. Computers were fun at first, but they are mostly boring now.

F. I like computers, but I learn most of what I want to know about computers outside of school.

# PARTICIPANTS IN THE DEVELOPMENT OF THE COMPUTER COMPETENCE OBJECTIVES

The National Assessment appreciates the efforts of all the individuals who contributed to the development, writing, and review of the 1985-86 objectives in computer competence. Many area specialists, educators, school administrators, legislators, members of the business community, parents, and lay people participated at various stages of the development process and this statement of objectives reflects a consensus of their views and opinions.

## LEARNING AREA COMMITTEE

Marc S. Tucker (Chairman), Carnegie Forum on Education and the Economy, Washington, DC
Harold Abelson, Massachusetts Institute of Technology, Cambridge, MA
Sharon Burrowes, Wooster Public Schools, Wooster, OH
Eugene Galanter, Columbia University, New York, NY
Vinetta Jones, University of North Carolina, Chapel Hill, NC
Arthur Luehrmann, Computer Literacy, Berkeley, CA
Michelle Rohr, Houston Public Schools, Houston, TX
Joyce Tobias, Brookline Public Schools, Lexington, MA

## COMPUTER COMPETENCE REVIEWERS

Leon Adkison, Taylor University, Upland, IN
John Aiken, The Evergreen State College, Olympia, WA
Judy Anderson, E. Consolidated School, St. Paul, MN
Edward Asmus, State University of New York at Buffalo, Buffalo, NY
Margaret Asprey, David Systems, Inc., Sunnyvale, CA
Richard Austing, University of Maryland, College Park, MD
George Bass, Jr., College of William and Mary, Williamsburg, VA
Henry Jay Becker, Johns Hopkins University, Baltimore, MD
Frank Besag, University of Wisconsin, Milwaukee, WI
Karen Billings, Houghton Mifflin, Boston, MA
Gary Bitter, Arizona State University, Tempe, AZ
Susan Brinkman, National PTA, Chicago, IL
Hugh Cobb, Consultant, Denver, CO
William W. Crandall, Princeton University, Princeton, NJ
Enrique Cubillos, Washington D.C. Public Schools, Washington, DC
John DeGilio, Vassar College, Poughkeepsie, NY
Tim Ellard, Opinion Research Corporation, San Francisco, CA
Gail Feldman, Parent, Braintree, MA
Robert Feldman, Parent, Braintree, MA
Mark Fetler, California State Dept. of Education, Sacramento, CA
LeRoy Finkel, San Mateo County Office of Education, Redwood City, CA
Glen Fisher, Computer Using Educators, San Jose, CA

Steven M. Frankel, Montgomery County Public Schools, Rockville, MD
Susan Friel, Lesley College, Cambridge, MA
Beverly Hagy, Trenton J.H.S. #5, Trenton, NJ
Sally Hansen, Mercer County Community College, Trenton, NJ
Dennis Hartzler, Oak Street Elementary School, Orville, OH
Marge Hohly, Downey High School, Long Beach, CA
Beverly Hunter, Targeted Learning Corporation, Amissville, VA
Elois Irvin, J.F. Kennedy High School, Richmond, CA
Robert B. Jensen, Farmbank, Inc., Aurora, CO
Scott Johnson, High School for the Engineering Professions, Houston, TX
Dan Kauffman, Florida Atlantic University, Boca Raton, FL
Bette Kindman-Koffler, Mathematica, Inc., Princeton Junction, NJ
Charlotte Kuh, AT&T, Short Hills, NJ
Rafael Ledesma, St. Joseph of the Palisades High School, West New York, NJ
Peter Lenkway, Department of Education, Tallahassee, FL
Thomas Lough, Piedmont Community College, Charlottesville, VA
Michael S. Mahoney, Princeton University, Princeton, NJ
David Martin, Jericho High School, Jericho, Long Island, NY
Mara Mayor, Annenberg - CPB Project, Washington, DC
Diane McEntyre, Mills College, Berkeley, CA
John McManus, Pepperdine University, Manhattan Beach, CA
Patty Mitchell, National Assoc. of State Boards of Education, Alexandria, VA
Michael Moshell, University of Tennessee, Knoxville, TN
David Moursund, University of Oregon, Eugene, OR
Adeline Naiman, HRM Software, Cambridge, MA
Robert Peebles, Superintendent of Schools, Alexandria, VA
Robert Perry, Santa Monica High School, Santa Monica, CA
Stan Pogrow, University of Arizona, Tuscon, AZ
James Poirot, North Texas State University, Denton, TX
Ruth Randall, Minnesota Department of Education, Apple Valley, MN
Jean Rice, Computer Education Consultant, Minneapolis, MN
Ellen Richman, Chagrin Falls Public Schools, Morland Hills, OH
Dick Ricketts, University of Oregon, Eugene, OR
Boots Riebling, St. Thomas Aquinas High School, Wilton Manor, FL
David C. Rine, Western Illinois University, Macomb, IL
Linda Roberts, Congress of the U.S., Washington, DC
Jean Rogers, University of Texas, Austin, TX
Randolph Ross, Newtown High School, Elmhurst, NY
Nola Sheffer, Hanscom Primary School, Hanscom AFB, MA
Joy Silver, Congressional Aide, Washington, DC
Twila Slesnick, Classroom Computer Learning, Belmont, CA
Sally Sloan, Minneapolis Public Schools, North Oaks, MN
Elliot Soloway, Yale University, New Haven, CT
Joan Targ, Interactive Sciences, Palo Alto, CA
Robert Taylor, Teacher's College, New York, NY
Robert F. Tinker, Technical Education Research Centers, Inc., West Concord, MA
Decker Walker, Stanford University, Stanford, CA
Linda Weinert, Versatech Inc., Santa Clara, CA
Judy Weisenberger, The Ohrenberger School, West Roxbury, MA
Leslie Wolfe, National Organization for Women LDEF, Washington, DC