

A COMPUTER-ASSISTED INSTRUCTION SOFTWARE PACKAGE
FOR COMPUTER LITERACY: DESIGN AND DEVELOPMENT

By

YIN-HWA CHOW

Bachelor of Law in Economics

University of Chinese Culture

Taipei, Taiwan

1978

Submitted to the Faculty of the Graduate College
of the Oklahoma State University
in partial fulfillment of the requirements
for the degree of
MASTER OF SCIENCE
July, 1988

Thesis
1988
C552c
cop. 2



A COMPUTER-ASSISTED INSTRUCTION SOFTWARE PACKAGE
FOR COMPUTER LITERACY: DESIGN AND DEVELOPMENT

Thesis Approved:

Richard Auberman

Thesis Adviser

Zane K. Zuehl

Dennis L. Mott

Norman D. Dunbar

Dean of the Graduate College

ACKNOWLEDGEMENTS

In the name of the God, the most beneficent, the most merciful. I thank the Lord whose worship makes me strong in faith, and whose obedience makes me closer to Him. It would have been impossible to complete this thesis without the support of God.

I would like to thank my master's program committee chairman, Dr. Zane K. Quible, for his encouragement, assistance, excellent, and constructive criticism that made my work satisfying and enjoyable. Thanks and appreciation is also given to the members of my thesis committee, Dr, Dennis L. Mott and Dr. Richard A. Aukerman, for their assistance, time and effort spent throughout my master program. I am especially appreciative to my thesis adviser, Dr. Richard A, Aukerman, who aided me in conducting this research. I wish to extend my appreciation to him for the excellent comments, suggestions, and close supervision.

Appreciation is expressed to my colleague, Mrs. Ruthann Dirks for her useful and stimulating discussion. I also grateful to Ms. Azizah Mohd.Noordin who allowed me to use her computer facilities for this study.

Finally, special thanks is given to my family for their love, confidence, support, and continual encouragement which was provided as I worked to finish this thesis.

TABLE OF CONTENTS

Chapter	Page
I. THE RESEARCH PROBLEM	1
Introduction	1
Statement of Problem	4
The Purpose of the Study	4
Need for the Study	6
Delimitations of the Study	10
II. REVIEW OF RELATED LITERATURE	11
History of Computers in Education	11
The Characteristics of Computer-Assisted Instruction	20
Factors of Success in Computer-Assisted Instruction	27
Current Status of Computers in Education ...	30
Trends of Computer-Assisted Instruction	33
III. METHODOLOGY	37
Drill and Practice	39
Tutorial	42
IV. RESEARCH DESIGN AND PROCEDURES	47
Review Related Literature	48
The Study Instrument	49
Determining the Content of Instruction	50
Designing and Laying Out the Instruction ...	51
Drill-and-Practice Design	55
Determining the Number of Questions	56
Determining the Number of Tries ...	57
The System Flow Chart	57
Detailed Flow Chart	58
Design and Layout of Frames	63
Lesson-Menu Frame	63
Introductory Frame	64
Question Frames	67
Entry-Error Frame	71

Score Frame	71
Ending-Menu Frame	71
Tutorial Design	75
Designing the System Flow Chart ...	75
Design and Layout of Frames	76
RULEG Frames	83
EGRUL Frames	83
Comparison Frames	84
Creating Graphics	84
Coding, Testing, and Debugging	95
Compiling and Linking	96
VI. SUMMARY AND RECOMMENDATIONS	98
Summary	98
Recommendations for the Future Research	104
BIBLIOGRAPHY	106
APPENDIX A - THE RESULT OF DRILL INSTRUCTION FOR SELECTED LESSON	111
APPENDIX B - THE RESULT OF TUTORIAL INSTRUCTION FOR SELECTED LESSONS	152

LIST OF FIGURES

Figure	Page
1. Drill and Practice Structure	40
2. Tutorial Structure	44
3. Organization of CAI Software	52
4. System Flow Chart of Drill-and-Practice	60
5. Detailed Flow Chart of Drill-and-Practice	61
6. Lesson-Menu Frame of Drill-and-Practice	64
7. Introductory Frame of CAI Software	65
8. Introductory Frame of Drill-and-Practice	66
9. Correct Response at First Try of Drill-and-Practice	68
10. Correct Response at Second Try of Drill-and-Practice	69
11. Incorrect Response at Both Tries of Drill-and-Practice	70
12. Entry-Error Frame of Drill-and-Practice	72
13. Score Frame of Drill-and-Practice	73
14. Ending-Menu Frame of Drill-and-Practice	74
15. Closing Frame of Drill-and-Practice	74
16. System Flow Chart of Tutorial	77
17. Lesson-Menu Frame of Tutorial	78
18. Introductory Frame of Tutorial	79
19. Entry-Error Frame of Tutorial	80
20. Correct Response Frame of Tutorial	81
21. Incorrect Response Frame of Tutorial	82

22.	RULEG Frame 1 of Tutorial	85
23.	RULEG Frame 2 of Tutorial	86
24.	EGRUL Frame 1 of Tutorial	87
25.	EGRUL Frame 2 of Tutorial	88
26.	EGRUL Frame 3 of Tutorial	89
27.	EGRUL Frame 4 of Tutorial	90
28.	Comparison Frame 1 of Tutorial	91
29.	Comparison Frame 2 of Tutorial	92
30.	Graphics Frame 1 of CAI Software	93
31.	Graphics Frame 2 of CAI Software	94

CHAPTER I

THE RESEARCH PROBLEM

Introduction

The computer revolution in the last two decades has greatly influenced the educational system in the United States. Both education and training systems have used the available computing resources for many purposes. The primary use of computers in educational institution is for computer literacy plus instruction in the fundamental concepts of computer operations and programming.

The U.S. has been transformed from an industrial society into an information society that relies heavily on computers and related information systems (Hepner,1982). Statistics have shown that the majority of employees work with information processing and communications. Computer technology has also played a major role in information processing and data communications. Thus, the recognition and understanding of the important features of computers is not only important, but essential to the American workforce. Unfortunately, the special features and the general limitations of computers are not commonly apparent by most students.

Recognition and understanding of the important features of computers in an information society has obligated government officials and educators to launch an educational effort to remove or greatly reduce computer illiteracy. It is important for students to at least understand the nature, features, and importance of computers in an advanced society. Therefore, it also appears important for teachers to have a working knowledge of computer literacy. When is operationally defined as the knowledge, skill, and attitudes needed to function efficiently and effectively in computer-related operations. Since the school is the main institution for socialization and training of citizens, computer literacy education should be provided to future citizens through the school system.

Computers usage and computer education implies a need for a broad spectrum of computer literacy education and training. Students need to know how to use the computer as a tool to process information, and they need to realize both the capabilities and the limitations of computers. The recognition and understanding of various features of computers helps students realize that computers do not have intelligence.

The objective of computer usage in education is often focused on teaching about the computer. While in industry training the focus is concentrated more on teaching with the computers. This process is referred to by a variety of

names, including:

CAI - Computer-assisted instruction

CBE - Computer-based education

CAL - Computer-assisted Learning

IAC - Instructional application of computers

CBI - Computer-based instruction

"CAI" is used in this study because it is based on instruction rather than education in general.

According to Orwig (1983), CAI is defined as instruction which meets the following two criteria:

1. The computer serves as the primary vehicle for the delivery of the instruction. What the computer actually delivers (the subject matter) can be almost anything. While many people associate CAI with math and science instruction, this is only because math and science teachers were among the first to experiment with CAI. Currently some of the most exciting CAI programs are in the areas of music education and creative writing.
2. The learner and the computer are in direct communication with each other. This process is called "interaction," and it places CAI into a category of instruction shared with human instructors. While the computer at its best is a poor substitute for effective human interaction, a well-written CAI program is capable of proceeding at a pace determined by the learner. It is also capable of modifying the presentation of the instruction to match the capabilities of the individual learner. Contrast this with other forms of instruction (a video tape, or perhaps even a classroom lecture) which do not involve interaction and that proceed at a fixed pace, with no consideration for the needs of individual learners.

An elementary student practicing a vocabulary game on the computer is directly involved with CAI. Similarly, an

astronaut is also involved in CAI when he/she is performing the landing on the surface of the moon through the use of computer simulation. A pilot can perform a hazard test on an aircraft simulator while performance of such a test is impractical for beginners with the actual aircraft. The requirements of using the computer as a vehicle and the needs of computer-interaction are clearly stated in the above cases.

Statement of the Problem

The problem of this study was to design and develop a CAI software package to assist in teaching a computer literacy course. The material consists of integrated computer literacy concepts and computer programming with the BASIC language.

Purpose and Objectives

A CAI software package can be used as a useful guidance aid for students who study about computer literacy. The package should also be useful to the individual teacher, student, industrial trainer, or computer professional who intends to develop an instructional computer program. This study had the following objectives:

1. Presenting information

An important means of presenting information is through example. In addition to stating the physical rule of "force

equals mass times acceleration," an instructor could demonstrate the application of that rule, such as a truck accelerating more quickly when it is empty. The demonstration of such applications is more feasible with CAI software.

2. Guiding the student

The second objective, guiding the student, is to guide the students through interactive processing. Students make errors and, at times, are not aware that they have made them. Teachers not only guide the students but also assist in correcting their errors. The interactive process of the student attempting to apply new knowledge, guided by the instructor, is an important component of instruction that is often omitted. The use of CAI makes this interactive process available.

3. Provide the chance for students to practice

The instructional process is not complete when the student does something once. The student must be able to perform quickly and fluently, sometimes under conditions of distraction, with few or no errors. Furthermore, practicing a skill a single time or answering a single question will not guarantee retention. Repeated practice through CAI makes the student more likely to retain information and increase fluency.

4. Assessing student learning

Tests provide information about the level of learning,

the quality of teaching, and future instructional needs. They should emphasize guiding instruction as well as assigning grades, in order to determine the students needs. Several drill problems and sample tests can be provided to the students by CAI. The computer will alert them when the required time has elapsed, as well as provide their final score. If the final score is above passing level, the student has demonstrated professioning for the lesson; if below passing, he/she can easily review the lesson.

Need for Study

Advanced technology and lower equipment cost will assist in bringing about a major revolution in the ways of learning. The revolution will occur within the next decade and will affect the educational system at all levels. Schools have placed an inordinate emphasis on teaching about computers because the actual hardware was not available. Today, however, classrooms full of computers are becoming much more common. As a result, schools will probably increase the use of computers for teaching and CAI can play a major role in education for and about computer.

The advantages of the computer as a learning and teaching device are the constant interaction and unique individualization. Each student at a computer display has a unique interactive experience based on the student's past performances or other needs based on teacher assessment.

Students can control the pace of learning, which is normally controlled by the teacher in a lecture situation. They can review the material at any desired pace and can be given more advanced material as appropriate. The major advantage of CAI is the transformation of learning into an active process, with students playing a constant thinking role. In a lecture, not all students actively participate in the learning process. Most educators agree that the active learning process is superior to passive learning. Thus, the combination of increasing educational effectiveness plus the decreasing cost of hardware will become the primary generator of an educational revolution based on computing.

Computer-assisted instruction will benefit both teachers and students when properly implemented. Unfortunately, at the present time, there is a critical shortage of software for CAI. According to Burke, publishers, computer software houses, and computer manufacturers had little interest in CAI as late as 1978, but now they are searching vigorously for CAI software. The reason for this demand stems from the development of inexpensive, powerful, and reliable microcomputers. The introduction and rapid spread of these microcomputers, which have several advantages over the mainframe computers, has dropped the average cost per terminal hour from several dollars per hour on each terminal of a large computer to a few cents per hour on a microcomputer. This factor increased the potential for CAI

users at an exponential rate. (Burke, 1982)

The rapid acceptance of microcomputers in schools and training organizations has created a tremendous demand for CAI. Initial applications are varied, from word processing to teaching computer programming. However, the natural progression is to use the microcomputer for CAI. CAI in this sense means direct instructional delivery in which the computer mediates instruction, simulating a human tutor.

At the 1981 Microcomputers in Education Conference, J. Edwards spoke on "The Courseware Crisis." She stated that a crucial obstacle to the wider use of the microcomputer is the lack of adequate material including lack of documentation and dissemination of the materials. This news did not surprise those educators who have tried to acquire software that is educationally suitable, relevant to their school's needs, and available at a reasonable price. According to Sadowski, the available software is often poorly written. Some school districts have ordered software and thrown away more than half of the disks after teachers have seen them once. Other school districts' administrators are concerned about the quality and effectiveness of the software and only purchase those software packages that are already approved by their teachers or upon the recommendation from other users (Sadowski, 1984).

Several attempts have been made at the national level to solve "The Courseware Crisis." Microcomputer software and

information for teachers (MSIFT) is a clearinghouse for microcomputer instructional software and teacher information which has been established for the collection, evaluation, and dissemination of material related to microcomputer software packages. Because of the increase in the number of software packages, MSIFT is not able to evaluate every software package.

The inherent problem of commercial software is the lack of instructional flexibility. For example, a teacher might want to modify or change a drill problem, or a new lesson might be desired. These minor modifications are not even possible with most commercial software because the program listing is not available. As a result, teachers are restricted to the available features that are often not desirable. Ideally, CAI software should be flexible to allow teachers to modify desirable aspects or eliminate the undesirable features.

Software limitations and other obstacles such as price, and lack of trained educators, create an environment for a crisis in courseware. The major crisis in the courseware can be resolved by teacher-made software. A solution is to find teacher who knows not only how to write computer programs but also who knows what can be effectively taught on a computer. By utilizing the teacher's skill in programming and instruction, a school can develop its own instructional software package which is tailored to the needs of the

district's teachers and students. A CAI package developed by an educator or group of educators can be more effective than any other commercial software that is designed by a group of programmers with lack of knowledge in education (Edward, 1984).

Lack of comprehensive CAI instructional materials plus materials that do not relate to present or new curricula and material designed for large systems which cannot be used directly on microcomputers create additional problems.

This study involves the design and development of a CAI software package to provide an efficient and effective instructional program for a course on computer literacy.

Delimitations of the study

The following delimitations were imposed in this study.

1. This study is delimited to the scope of computer literacy.
2. This study is further delimited to BASIC programming and basic knowledge of the computer.
3. The software package is delimited to the IBM PC and IBM PC compatible computers. It is written for a color monitor with graphics capability, but will work on a monochrome monitor with a graphics adaptor.

CHAPTER II

REVIEW OF RELATED LITERATURE

This study concerns the design and development of a CAI software package for a course on computer literacy. The related research papers and literature were reviewed with concentration on the following topics:

- (1) The History of Computers in Education
- (2) The Characteristics of Computer-Assisted Instruction
- (3) The Factors of Success in Design and Development of Computer-Assisted Instruction
- (4) The Current Status of Computer in Education
- (5) Trends of Computer-Assisted Instruction

History of Computers in Education

The first instructional application of the computer started in 1950 when the scientists at the Massachusetts Institute of Technology (MIT) introduced the specialized electronic computer, the Wirlwind. The Wirlwind, which was designed originally as a flight simulator for training combat pilots, occupied three floors because of large size and high power consumption of the vacuum tubes (Hofmeister, 1984). The invention of the transistor in 1948 made the second

generation computers more powerful and practical because of their dependability, small size, and low power requirement.

The first use of computers by educational institutions was at the end of the 1950s when large universities began to use computers for administrative purposes, such as accounting, payroll, and student record keeping (Alessi and Trollip, 1985). At the same time research centers began to use computers for their instructional research. Educators had considerable difficulty working with the early computer systems because the courseware had to be written in a complex computer language. The introduction of authoring languages allowed the educators who had no knowledge of the complex computer languages to develop courseware. The PLATO project at the University of Illinois was one of the research applications which began in 1960 with the goal of designing a large computer-assisted instruction. The authoring language developed in association with the PLATO project was called "Tutor" (Hallworth and Brebner, 1980).

In the early 1960s, IBM researchers introduced an authoring language entitled "Coursewriter." The Coursewriter was developed for preparing instructional materials on IBM large computers. Both the Tutor and Coursewriter languages went through many revisions. These two authoring languages were among the best quality computer-assisted instruction courseware written for the main frame system. The widespread application of microcomputers in education required the

design of authoring languages for the personal computers. Pilot became the authoring language most widely used on microcomputers. It was developed in 1969 by John Starkweather at the University of California medical center. The development of general purpose computer languages made the task of instructional programming easy, efficient, and effective. In 1963, the most general purpose computer language developed for the microcomputers was Beginners All-Purpose Symbolic Instruction Code (BASIC). The introduction of BASIC had a major role in development of instructional programming (Hofmeister, 1984).

In 1965, the Learning Research and Development Center (LRDC) at the University of Pittsburgh had a computer system that provided spelling drills as part of their investigations to use computers for educational research. The LRDC equipment included a Digital Equipment Corporation PDP-7 computer costing approximately \$50,000 (Ragsdale, 1982). The introduction of inexpensive microcomputers brought a major change in price of spelling drill facilities. In 1982, Texas Instruments developed a spelling drill device that incorporated a microprocessor, keyboard, and random-access audio at a price of approximately \$50. Up to 1982, due to expensive costs associated with CAI on large computer systems, there had been little practice with CAI in most schools.

By 1964, integrated circuits (ICs) had brought a new

revolution in computer technology, and much of the second-generation transistor technology had been replaced with ICs. The ICs have the transistor functions built into miniature circuits in chips of silicon. Computers heavily dependent on IC technology rather than transistor technology were known as third-generation computers. Third-generation computer technology has gone through several levels of miniaturization of circuitry.

From 1965-1971, the third generation computers became available in increasing numbers and at lower cost. Many schools began using computers for administrative functions, but the instructional use of computers was still mainly for private corporations, military offices, and large universities. The first CAI software package was developed by the researchers of computer companies to train their staffs. In 1959 researchers at IBM's Watson Research Center took their staff development CAI technology and used it with school children in the state of New York (Kinzer, Sherwood, Bransford, 1986). Several teletype terminals were placed in the public schools and they were connected to the IBM computer through telephone lines. After the early work by the staff at Watson Research Center, two universities established CAI research and development efforts to give direction to the future work of researchers developing CAI software and instructional programming. In 1963, Stanford University established a research center for design and

development of programs in computer-assisted instruction. This project and the PLATO project at the University of Illinois made major contributions in the development of courseware for large computer systems. Their efforts also lead to the development of hardware and software more compatible with the needs of the public schools (Callison, 1985).

The efforts of Donald Bitzer and his PLATO team at Stanford University generated widespread interest in the application of computers to instruction. Their effort also helped to grow awareness of the potential for computers in society. In 1967, computer literacy for school children was recommended by the President's Science Advisory Committee. This recommendation was one of the earliest indicators that showed computers can be used for more than just an administrative and instructional tool in the schools (Hofmeister, 1984).

In 1967, a group of Minnesota school districts formed an association entitled Total Information for Education Systems (TIES). Minnesota was one of the most progressive states in the application of computers to both the business and instructional aspects of education. By the end of the 1960s many private corporations showed an interest in computer-assisted instruction. IBM continued to develop its CAI software and hardware for training of its own staff. Most of the CAI courseware were designed for the large and expensive

main frame systems. Many of these computers and existing courseware were designed for business applications and were not well suited for computer-assisted instruction (Ragsdale, 1982)

In 1968, minicomputers were introduced. They were lower in cost than the large main frame system but had excellent facilities well suited for computer-assisted instruction. Although the minicomputers had reduced the costs and increased the reliability of computer services, the minicomputers were still relatively expensive and the CAI courseware suited for these devices was limited. The IBM 1500 designed in 1966 was the first computer system designed for CAI (Merrill, Tolman, Christensen, Hammons, Vincent, Reynolds, 1986).

In 1972, the MITRE corporation and Brigham Young University started development of the Time-Shared Interactive Computer Controlled Instructional Television (TICCIT). Courseware for the TICCIT system was developed first at Austin, Texas, and then later at Brigham Young University in Utah. The TICCIT facilities helped students to study lessons presented on standard color televisions and interact through modified typewriter keyboards. All of these interactions were controlled by a minicomputer. The TICCIT system followed after earlier development of the IBM 1500 for educational applications. TICCIT was not a brand new computer but only a combination of a few existing components

in a unique configuration specially designed for instructional applications (Merrill, Schneider, Fletcher, 1980).

The cost of the TICCIT system was considerably reduced because its components came from the computer and television industries. A number of specialized teaching strategies were developed as a part of the TICCIT system. Additional keys also were added to the standard keyboard. The keyboard in the TICCIT system had special keys labeled, Rule, Example, Practice, Easy, Hard, and Advice, in addition to the ordinary alphanumeric keys. By pressing these keys the students could change instructional activities. They could ask for easier or harder activities or ask for some advice as what to do. The TICCIT was notable not only for its creative integration of hardware but also for its sophisticated operating software. By using TICCIT, a wide range of instructional programming was possible through the combination of computer and videotape technology (Kinzer, Sherwood, Bransford, 1986).

In the early 1970s the PLATO IV, a large time-shared instructional system, was introduced. The PLATO IV was used to provide on-line lessons to hundreds of students. All program execution occurred on the main computer. Students studied on individual terminals and could communicate with the main computer through telephone lines. PLATO IV allowed up to 600 students to use the computer simultaneously (Allessi and Trollip, 1985). In addition, the authors could

develop instructional programming at the same time that students were studying lessons.

Other CAI projects were developed at MIT in the early 1970s (Papert, 1971). Papert began research on teaching children by having them program computers. Two major developments of Papert's project are LOGO and TURTLE. LOGO is a powerful but easily learned programming language, and the TURTLE is a small robot that children can control using LOGO.

In 1972, the National Science Foundation urged educators to start planning for the computer-literacy of school-age children. Interest in computer applications in the instruction of school-age children and education continue to grow. The Minnesota Educational Computer Consortium (MECC) was formed in 1973. The goal of the MECC project was to put computers and all related facilities in the public schools. The MECC also used large and relatively expensive computers (Anderson, Klassen, 1981).

In the mid 1970's, a few small computer companies started to experiment with microcomputers, but it was not until 1977 that microcomputers with serious educational applications appeared. The low cost, ease of use, portability, and the general purpose nature of these microcomputers excited a wide range of educators. Two large corporations, Radio shack and Commodore Business Machines, introduced the TRS-80 and PET computer, respectively. The

Apple computer company introduced the Apple computer with higher quality than TRS-80 and PET. (Alessi, Trollip, 1985)

Up to 1977, all projects in computer-based instruction or computer application in business were supported for large mainframe systems. By the introduction of powerful and low cost microcomputers, it became possible for the individual university researcher, the small public school, and public school teachers to buy the microcomputer and to begin developing their own CAI software package. From 1977 to today, the growth in the educational use of computers in the schools and colleges is seen. All colleges and universities have microcomputers, and most elementary and secondary schools have them, too.

Most of the large investments in high quality courseware were in projects such as the PLATO project designed for large main frame systems. This courseware was not available for the thousands of microcomputers in use in the schools. Therefore, teachers were making CAI programs in their free time. Computer programmers were working to develop courseware for this rapid growth market. Since some teachers may not have good experience with computer programming, and some computer programmers may not fully understand the content of the required courseware, some courseware was of high quality, but much of it was poorly conceptualized, limited in its operation, and fragmented in its structure (Alessi, Trollip, 1985). Thus, an educator with good

experience in computer programming can develop high quality courseware which can be used in many school districts.

The Characteristics of Computer-Assisted Instruction

The most obvious difference between traditional classroom instruction and CAI is the human interaction in CAI with a machine rather than with another human being (Steinberg, 1984). The interactive capability of CAI is one of its most important instructional characteristics. In the regular classroom, only one person can answer each question posed by the teacher, while others are listening. But in CAI, every student can respond to every question that has been raised.

A well-written CAI program can react to each response with an individualized message. This message is called "feedback" and can be far from a logical expression such as "true" or "wrong", and "good" or "bad." The feedback can inform the student why the response is wrong plus showing the correct response. Assume the question is "What is the ROM"? If the student's response is "Random Only Memory," the computer may reply, "Sorry, Random Only Memory is not a standard computer term; the correct answer is Read Only Memory." In addition, in CAI students answer the questions in the privacy of their interaction with the computer terminal.

In order to provide individualized interaction between

the student and the computer, the CAI software must have a number of decisions and plans for the student's response. Well-written CAI courseware should have the following capabilities: (1) choosing proper conditions for responding, (2) guiding the students to enter responses, and (3) judging student's responses (Burke, 1982).

A serious problem in CAI software packages is that computers may not be able to understand human speech. The recognition and understanding of human speech by computers has been studied extensively by several researchers, and the results are promising. In the classroom, the students may answer the questions in many different ways but still expect the instructor to understand them. In CAI, the answer must be given exactly as it was designed because computers are not capable of understanding the natural languages.

CAI is designed for use by a single person at a time. It is normally used by only one person and usually operates on only one program at a time (Orwig, 1983). This characteristic of CAI is referred to as individualized pacing.

An individualized program is written to work with one person at a time and always considers the student's learning characteristics. In group instruction, a student who does not understand a concept must move along with the rest of the class although he/she may not be ready to do so. In CAI, each student can work at an appropriate level of difficulty

and proceed in the lesson at an appropriate pace. The extra drill problems and practices are used to help students understand new concepts. A student who learns quickly does not have to repeat the materials that he or she has already learned. This individualized instruction is the remarkable characteristic of a high quality CAI program.

Since the computer is capable of quickly evaluating the answers provided by the users, the immediate feedback will promote the students to learn. A well-written CAI courseware can praise the user for correct answers, correct him or her in the case of a wrong answer, and even recommend an alternative learning strategy (Orwig, 1983). In addition to evaluating the responses for providing immediate feedback, the computer can store the student's records for future evaluation. This record can be used by the instructor in keeping track of student progress, by the user as a review lesson, and possibly by the computer to start the user at the beginning of the next lesson.

Several researchers have found that augmenting classroom instruction with CAI courseware provides superior performance on the Scholastic Aptitude Test (SAT) (Grady and Gawronski, 1983). Other reviewers (Burs and Culp, 1980 and Johnson and Jongejan, 1981) have reported that supplementary instruction with CAI leads to a higher achievement and the amount of time required for learning mathematics and language arts skills is significantly reduced. Johnson and Jongejan (1981) found

that instructional activities delivered by a computer required less time to complete than any other means. They also reported the effects of drill and practice with CAI and using computers to present questions and review and evaluate a response has generally been successful. The majority of these studies showed that using CAI courseware instead of traditional classroom instruction provides a saving in time to complete a course of study, greater efficiency in achievement per unit of time, and improvement in skills.

It is known that CAI courseware involves the individual actively in the learning process. It is almost impossible for the student to be a totally passive member while using CAI. This activity and continual involvement and participation facilitate learning as students progress through the instructional material (Walker and Hess, 1984). Reinforcement of learning in this situation is immediate and systematized which should result in more effective learning.

The effectiveness of CAI has been defined differently by many investigators. To some, effectiveness means the amount of time required to complete one lesson compared with the time required to complete the same lesson using traditional instruction. To others, it means the degree of learning, or whether or not the student stays in or drops out of a learning experience. Some are simply concerned with the student's change in attitude toward the computer as an instructional medium. Levien (1972) reported that a CAI

physics course developed at Florida State University was very effective. Tentative evaluation indicated that overall instructional time was reduced by 17 percent over the traditional lecture course. In addition the students scored higher on final exams and attained superior knowledge of the physics course. Stanford and the University of Southern California have used several computer-assisted courses to teach their students (Levien, 1972). Studies of these CAI coursewares revealed positive results in terms of student performance on examinations, student behavior, and the student responses to a questionnaire about the program. In addition, fewer students dropped out of the computer-based courses.

Chambers and Sprecher (1980) found that low aptitude students profit more from CAI than those with average or high aptitude. CAI can have substantial impact on instructional practices of many students in a special classroom and especially can help those students who have considerable difficulty in the learning process. The computer is an infinitely patient teacher, providing experience and reinforcement as they are needed. It can be used for instructional purposes and as an alternative means for special needs persons to minimize or counteract the effect of a disabling condition. A well-written CAI, dealing with perceptual difficulties, and with the needs of blind, hearing-impaired, nonverbal, and deaf students, can have an

incredible impact on the learning of the special students. CAI can be used as a flexible, portable, audiovisual aid to provide a better illustration of ideas.

A review of the literature reveals that CAI has several advantages over traditional education. Some of these include:

1 . Provide students with optional ways to learn

Through using the CAI, the teacher can provide appropriate educational opportunities to all students, including both the academically talented and those who have difficulty in the learning process.

2. Provide individualized instruction

CAI is designed for use by a single person at a time. Each student can work at his or her own pace. The computer is suitable for students of all achievement levels. It is also effective with both individual and small group learners.

3. Facilitate an infinite practice

If it is necessary to provide the student with a hundred exercises of the same type, the computer will proceed to do so. The responses are carefully evaluated and appropriate comments are given. Also, the computer, if correctly programmed, does not make any mistakes.

4. Provide drill for a large number of students

Through time sharing, a large number of students can access a computer simultaneously and work on the

same drill problem. Each student can proceed at his or her own speed at the same time.

5. Privacy of student-computer relationship

The completely impersonal relationship between the student and the computer is a distinct advantage. The student can make any mistakes in private without having to demonstrate his failure to the teacher and to the rest of the students.

6. Reduce the time for learning

Studies have shown that students need less time for learning through the computer.

7. Ease of updating

Many topics that are taught today are continually changing. A teacher-built CAI can be easily updated with new content. It offers an excellent adaptation to the special needs of handicapped students.

8. Provide immediate feedback

The computer is capable of quickly evaluating the response and providing the result. Thus the student can receive almost immediate feedback to his or her answer. This immediate feedback motivates the student and can be very beneficial.

9. Provide interactive capability

Students who use CAI can communicate with the computer interactively. The interactive facility provided by CAI enables the students to watch their scores and see the

general progress.

10. Lack of discrimination

The computer facilitates teaching methods to all students at all times and does not discriminate on the basis of race, sex, nationality, personality, background, or the previous day's behavior.

The Factors of Success in Computer-Assisted Instruction

The factors affecting the success of CAI in education depend on the motivational process in learning and the social structure of the educational system. Many researchers have tried to specify several design principles for using computers successfully in different educational environments. These principles that are actually useful in designing instructional environments seem to be the most obvious ones, but most often are violated.

There are several features that make the learning process enjoyable. Malone (1984) believes that these features fall into three categories, challenge, fantasy, and curiosity. CAI can be challenging if it includes a goal whose outcome is uncertain. One way of making outcomes uncertain is providing variable levels of difficulty that are either chosen by the player or determined automatically. Having a number of goals at different levels can often be encouraged by score-keeping or speeded response.

Fantasies in CAI software can make the activities

emotionally appealing (Malone, 1984). For example, if the student's response is correct, a pleasant music followed by a reward will motivate the student and encourage him/her to further practice. Similarly, if the student's response is incorrect, the computer feedback is displayed to inform the student about the need the second trial. The instructor/designer should pick fantasies very carefully in order to avoid any possible discouragement if the student's response is incorrect. The educational activities can be very effective if the audio and visual facilities, such as music and graphics, are used to lead learners into situations in which they are surprised. The surprising situations often are used to provide some information that helps the student to understand the unfamiliar concepts that led them to be curious in the first place.

Lesgold (1984) stated that massive amounts of drill and practice are needed to obtain basic skills. The practice must be motivating, appropriate, and capable of providing immediate feedback. In computer-based practice, every error in performance should be taken into account. The CAI system can be designed so as to increase the mental discipline and self-motivation of students.

Dugdale(1984) indicated that the material used for CAI should be designed to engage the students in productive thought and activity. The material should keep the student interacting and participating rather than passively watching

and listening.

Kosel (1984) has suggested a number of principles that are effective in the design of an instructional program. He believed that a crowded screen should be avoided. In addition, the users should be able to go through reading and standard timing loops must not be used. It is also important to provide graphics, color, and sound to enhance learning. The materials used in CAI software should be appropriate and fit different classroom situations.

Potts (1984) has stated that faculty support is the most important factor of success in CAI courseware. But the greatest problem in designing a CAI program is the financial one. This problem often can be solved by external resources received from industry and government for the funding of costly new programs. Academic computer directors and CAI administrators believed that faculty participation and support plays an important role in the design and development of a successful CAI program. Unfortunately, the CAI program has been ignored by many educators and the administrators themselves are not fully informed about its capabilities.

While some conservative educators are not ready to adapt the non-traditional instructional methods, some active educators are seeking new methods to improve the teaching and learning process. CAI is the best choice of instructional software designed for innovative and concerned educators.

One study has shown (Potts, 1980) that faculty members

and computer specialists are most often responsible for introducing the CAI program. If the school administrators offer some incentive to faculty members, a successful CAI program can be developed. The CAI software produced within the school by an educator or a group of educators can be more effective than one purchased from a commercial company or acquired from another college or university (Backer, 1986).

In the majority of institutions that produced their own CAI, the administrators' support greatly influenced the team development approach. The dedication and support of computer center personnel is also as important as administrator support in the success of CAI. However, the success of the final product of CAI courseware depends on the creativity of the individual author and the ways in which the subject matter is interpreted on the screen.

The Current Status of Computers in Education

In the mid-1980's, Nebraska, Pennsylvania, and Colorado promised to put state funds behind educational technology. Many people at the Colorado Department of Education worked very hard to integrate technology into the curriculum. Later, all other states along with the District of Columbia started to show their commitment and enthusiasm for using computers in school districts (Reinhold, 1986).

All states started to hire many people to coordinate their efforts for collecting information concerning their state

spending on computer technology, brands of the computers being used, and the ratio of students to computers.

According to Electronic Learning's 1986 Annual Survey, all states but two were able to report the brands of computers being used in their school districts and the amount of money spent for both hardware and software. Only ten states did not report their student-computer ratio. All states but two reported strong local interest in computer education.

This report clearly indicates that states are very enthusiastic about using computers in their curriculum. Recently, governors of several states have become involved in educational technology. For the first time, the National Governor's Association had a task force on educational technology. The governors of New Hampshire, Texas, Alaska, Hawaii, New York, Oregon, Virginia, and Puerto Rico made strong recommendations for increasing funding and support for educational technology (Reinhold, 1986).

The field of computer education needs a tremendous amount of funding and support for spending toward hardware and software facilities. During 1985-1986, an estimate of \$550 million for hardware and about \$130 million for software was spent. More money is also required for the personnel to run and manage computer programs. Since commercial software is not usually suitable for CAI courseware, states need to spend more money on evaluation efforts and research and development for creating their own software by a group of

educators. In 1986-1987, 39 states spent more money on software than they did in 1985-1986. Only 20 states reported they spent more on hardware in 1986-1987. This statistic indicates that funds are available for the research and development of good CAI courseware. School administrators, computer directors, and educators are responsible to take advantage of this opportunity and develop CAI courseware that is suitable for their school districts and that fulfills the needs of their instructors and students.

Several states have realized that appropriate software and training are essential if the computer is to be used effectively. About half of the states have focused their efforts on looking for the best software to satisfy their curriculum needs. Forty-three states have hired a computer coordinator -- someone whose primary responsibility was coordinating computer use for one school or for many schools. In addition to state computer coordinators, more than 38 percent of the U.S. school districts also have hired a part-time or full-time person coordinating computer use. It is worth mentioning that most state funds are going toward research, evaluation, and distribution to find out the best method of delivering computer education.

The Governor's Technology Task Force said that ten states already are working together to establish a satellite transmission and cable television network to provide courses in remote areas. Several states including Alaska, Virginia,

Maryland, and Texas reported that they have used these technologies to teach particular courses or curriculum in outlying areas where students may not have access to teachers to teach these courses.

In a national survey, Becker (1986) was interested in finding what the students are doing with the computers. In completing this survey, each primary computer-using teacher was asked to divide the student's computer use into five basic activities:

1. computer-assisted instruction
2. programming
3. word processing
4. discovery learning and problem solving
5. other activities.

He combined the elementary and secondary school responses together. These responses counted the same regardless of how much computers were used. He found that CAI constitutes about half of the computer activities. He also found that a majority of teachers feel that learning about computer programming and computer literacy are the most valuable functions of computers; and a majority of teachers in elementary schools feel that CAI is the best way to use computers.

Trends of Computer-Assisted Instruction

Educators are extremely optimistic about the future of

CAI. Their optimism is based on the rapid growth of computer technology, which has remarkably reduced the cost of computer-based facilities. The technological trends toward miniaturization and the psychological and social trends toward the use of computers in education will lead to significant developments by the year 2000.

In the year 2000, all students in all of U.S academic institutions will take advantage of advanced, inexpensive sophisticated microcomputers as powerful educational tools. Each student will be able to receive information or share resources and programs from all over the world without concern of equipment or location. The fast-growing electronic and digital equipment will make it possible to launch a communication satellite totally devoted to science and education (Atkinson, 1978). This educational satellite will eliminate distance as a barrier to access and use of CAI programs developed around the world. Each student will be able to have a computer-based video device which will be capable of presenting interactive programs. In addition, this video device will be controlled for various reading levels and will be automatically adaptive to student performance.

Since the microcomputers will be the most effective tool for educational purposes, the skills to study and develop an intelligent knowledge-based instructional system will be almost mandatory. The knowledge-based systems will

be developed to assist the students to recognize, articulate, and use diverse forms of information from history to mathematics.

At the present time, the minicomputers reach the market with massive amounts of software for word processing and scientific research. However, a good CAI software package to assist the student in learning is still not developed. By the year 2000, this difficulty will be overcome and the minicomputer systems with several microprogrammable chips containing compilers for several authoring languages will be marketed (Attala, 1978). While the advanced technology will make it possible to provide denser memories at a lower cost per bit, the design and development of a reliable and complete CAI software package for minicomputers will be mandatory. By the year 2000, the utilization of CAI on an around-the-globe basis is a dream of the educators (Hirschbuhl, 1978).

The implementation of CAI in Europe is currently tied to a weak and variable communication system. But the problems involved in utilization of CAI in Asia are primarily the lack of trained personnel and the lack of advanced technology required for development of a CAI system. Both Asia and Europe will solve their technical problems probably prior to the year 2000 (Walker, 1984).

Throughout the world CAI will be delivered by means of distributed networks that utilize a massive amount of memory

for the storage of the computerized courses having motion, color, and audio capabilities. Each minicomputer will communicate to the extensive libraries throughout its local network. This type of communication will reduce the load on minicomputers and enable the student to obtain diverse information from around the world almost instantaneously (McGinty, Tony, and Sally, 1987).

This world-wide development will do much to advance the CAI system and make it possible to teach the areas that are currently difficult to teach, hard to visualize, or almost impossible to understand with the available instructional systems. This means that major utilization of CAI will be at preschool levels and at the end of professional development as opposed to current usage by elementary and secondary students.

The major factors that will make the trends of CAI possible are authors who can develop sophisticated courses, the fast growing computer technology, linking around-the-globe networks by means of satellite communication, and the faith and dedication of the educators who are working to improve the CAI courseware.

CHAPTER III

METHODOLOGY

A CAI software package can be used as a useful guidance aid for students who study about computer literacy.

The package should also be useful to the individual teacher, student, industrial trainer, or computer professional who intends to develop an instructional computer program. This study had the following objectives:

1. Presenting information

An important means of presenting information is through example. In addition to stating the physical rule of "force equals mass times acceleration," an instructor could demonstrate the application of that rule, such as a truck accelerating more quickly when it is empty. The demonstration of such applications is more feasible with CAI software.

2. Guiding the student

The second objective, guiding the student, is to guide the students through interactive processing. Students make errors and, at times, are not aware that they have made them. Teachers not only guide the students but also assist in correcting their errors. The interactive process of the

student attempting to apply new knowledge, guided by the instructor, is an important component of instruction that is often omitted. The use of CAI makes this interactive process available.

3. Provide the chance for students to practice

The instructional process is not complete when the student does something once. The student must be able to perform quickly and fluently, sometimes under conditions of distraction, with few or no errors. Furthermore, practicing a skill a single time or answering a single question will not guarantee retention. Repeated practice through CAI makes the student more likely to retain information and increase fluency.

4. Assessing student learning

Tests provide information about the level of learning, the quality of teaching, and future instructional needs. They should emphasize guiding instruction as well as assigning grades, in order to determine the students needs. Several drill problems and sample tests can be provided to the students by CAI. The computer will alert them when the required time has elapsed, as well as provide their final score. If the final score is above passing level, the student has demonstrated professioning for the lesson; if below passing, he/she can easily review the lesson.

In order to satisfy the purpose and objectives of this study, the following two methods were used,

1. Drill and Practice
2. Tutorials

Drill and Practice

Drill and practice instructional software are exercises that help students maintain or strengthen their knowledge. This software provides an opportunity for students to come in contact with facts, relationships, problems, scientific concepts and principles, and vocabulary until the materials are committed to memory or some special skill is obtained. The best drill and practice software possesses an interesting format which students are encouraged to reuse for establishment of the stimulus-response association required for memorization of certain relationships (Bitter and Camuse, 1984).

Drill and practice has been found to be successful with simple problem solving, such as arithmetic facts, and complex problem solving, such as physical and social sciences. They also can be applied to simple paired-associate learning, such as spelling or foreign language word translation, and verbal information, such as definitions, historical facts, and vocabulary development. Those who are learning English as a Second Language (ESL) can use computer software to develop their vocabulary and strengthen association between corresponding words, expression, and grammatical constructions. Many software packages are able to use sound

and pictures to illustrate words, phrases, and sentences. The ESL students can use a minicomputer to communicate with teachers or classmates much easier than in a regular classroom. They can benefit from computerized, individualized instruction to perform exercises and to check their answers immediately, while textbooks cannot provide such feedback to students.

In drill and practice, emphasis is placed on practicing rather than guiding, but with the assumption that the student is already familiar with the information to some level. Drills must be preceded by instructional methodologies that present the information or with an appropriate simulation or tutorial. The function of drills is to provide practice and enable the student to review some facts, relationships, and concepts, assuming that initial presentation and guidance have been already given. The structures and procedures of drill and practice used in this study can be best illustrated by figure 1.

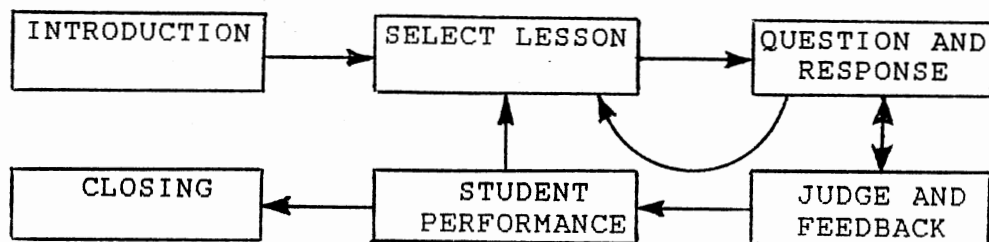


Figure 1. Structure of Drill and Practice

This structure begins with an introductory section that includes title and instructions. Following the introductory section, a cycle begins that is repeated many times. Each time the cycle is repeated, the following actions take place:

Alternative lessons are displayed.

A lesson is selected by the student.

A question is displayed.

The student responds.

The program judges the response and provides feedback to the student.

The questions continue to display until all of them are finished.

A summary of the student performance is given.

The cycle repeats as many times as the student wants, and the program can be terminated upon student request. Following the program termination, closing remarks are provided.

The design and development of drill and practice used in this study is based on the following principles:

1. Provide complete and clear instructions.
2. Provide a listing of the selected lessons for the student choice.
3. Use graphics, color, and music to motivate students and prevent boredom.
4. To personalize the program, call the student by name.
5. Increasing difficulty of lessons progressively,

starting out with easy items, and as the student masters them, choosing more difficult lessons.

6. Question, response, and feedback should be displayed in the same frame.
7. Ask frequent questions, especially comprehensive questions.
8. Allow students to answer a question more than once.
9. Give a short affirmation for the correct answer.
10. Give corrective feedback and remediation for the incorrect answer.
11. Keep feedback short and positive.
12. Feedback should be corrective, positive, diagnostic, reinforcing, informative, and prescriptive.
13. Make the end obvious with a short final message
14. When a lesson is finished, provide options for the student to exit, review the same lesson, or try another.
15. Allow the student to exit at any time based on the student request, and allow restarting.

Tutorials

Tutorial software utilizes written explanations, descriptions, questions, examples, and pictures for guiding the student and presenting the information. Tutorials often provide computerized software to determine the most appropriate beginning lessons for a particular student or

whether certain lessons should be skipped (Bitter and Camuse, 1984). The drill and practice often are presented after the tutorial software is introduced. Courseware which uses a tutorial software requires more complicated design and programming techniques.

Good tutorial software should be able to predict all possible correct responses and allow for insignificant errors such as misspellings and capitalization. In addition, tutorials should include both presentation and guidance. Computerized tests used in tutorials must be valid and reliable so that students are not subjected to already known material. These tests are used to evaluate learning fairly and accurately.

Tutorials are used in almost every subject area from humanities to the social and physical sciences (Alessi and Trollip, 1985). They are appropriate to present factual information; and assist in the way of learn rules and principles, or problem-solving techniques. Several different types of tutorial software exists. Some tutorials will present a series of screen displays to all users, regardless of individual differences among the students. Other types do not require all users to follow the same path, but will direct students to certain lessons or parts of a lesson according to the computerized test results. A well-written tutorial will present the material in a sequential fashion, which does not confuse the learner, and uses graphics and

sound to improve the instruction. In addition, student progress can be measured accurately, and he or she can control the pace of presentation.

The structure and sequence of the tutorial used in this study is shown in figure 2.

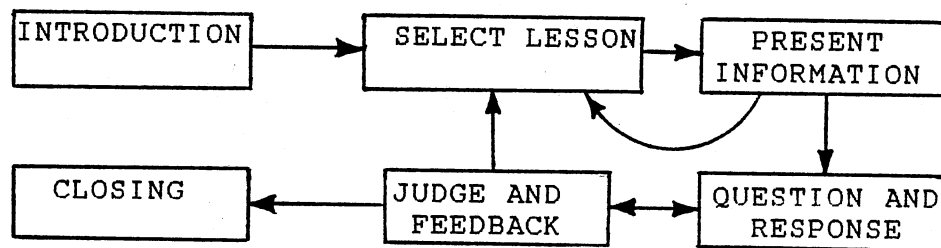


Figure 2. Structure of Tutorials

The structure of tutorials begins with an introduction that includes title and instruction. Following the introduction section, a cycle begins that is repeated many times. Each time the cycle is repeated, the following actions take place:

Alternative lessons are displayed.

A lesson is selected by the student.

Information is presented by means of examples and comparison of the outputs.

A question is displayed and the student responds.

The program judges the response and provides feedback or

remediation

The questions continue to display until all of them are finished.

The cycle repeats as many times as the student wants, and the program can be terminated upon student request. Following the program termination, closing remarks are provided.

The design and development of the tutorial used in this study is based on the following principles:

1. Use a short title page.
2. Provide complete and clear instructions.
3. To personalize the program, call the student by name.
4. Use color to emphasize the important information.
5. Stress clear transitions between presentations on different statements or lessons.
6. Use appropriate organizational methods for verbal information, concepts, rules and principles, and skills.
7. Layout should be attractive and consistent.
8. Provide remediation for incorrect answers.
9. Text should avoid scrolling.
10. Text should be clear and concise.
11. Make the end obvious with a short final message.
12. Allow the student to exit at any time based on the student request and allow restarting.
13. When a lesson is finished, provide options for

the student to exit, review the same lesson,
or try another one.

14. Feedback should be corrective, positive, supportive,
and immediate.

CHAPTER IV

DESIGN AND PROCEDURES

The following steps were used in researching the problem, planning the study, conducting the survey of Computer-Assisted Instruction, designing the CAI software package on Computer Literacy, and presenting the results of the study as a software package:

1. Review of related literature
2. The study instrument
3. Determining the content of instruction
4. Designing and laying out the instruction
5. Creating graphics
6. Coding, testing, and debugging
7. Compiling and linking

This study was designed to develop a CAI software package to assist in the teaching of computer literacy. It also can provide an alternative method of learning for the students. The CAI software package is designed to stimulate student interest in computer literacy. This software satisfies the four objectives that are required to establish well-written CAI courseware: (1) presenting information, (2) guiding the students, (3) providing a chance for the student to practice,

(4) assessing student learning. In addition, this software provides individualization and interaction. The students communicate with the computer interactively and can practice their lessons in a private environment. The color, sound, and graphic facilities also are used to enhance motivation and prevent boredom during the learning process.

Chapter IV describes the research design for gathering the background information, for study instrument used to select the proper equipment, for determining the procedure used to select the desired instruction, and for selecting the various analyses needed to fulfill the purpose of this study.

Review of Related Literature

The available professional publications and literature relating to computer-assisted instruction in education, CAI projects, and learning systems were examined. This study complements the background information related to CAI software and determines if a similar study has been made. The sources of information were the Educational Resources Information Center (ERIC) Research, an on-line search of a legal database by the Oklahoma State University library, Business Education Index, Business Periodical Index, on-line catalog by the Purdue University Library, numerous professional journals, computer magazines, and textbooks of computer literacy.

The literature was examined from the mid-1970's to the present (1987). The primary interest in the literature was on publications since 1980 when successful CAI software was designed and the latest information on CAI software was available.

The Study Instrument

This CAI software package is designed to run on the IBM PC and all compatibles with a Dos 2.1 or later version. The source code is written in the BASIC programming language using Advanced BASIC facilities. While this software can be run on a monochrome monitor with graphics adapter, but a color monitor with graphics adapter is preferred to see the high resolution color pictures used for motivation. The software is stored on a single 5 1/4" double side/double density diskette with executable format. An IBM PC or compatible with single disk drive and minimum 64 KB of program memory is sufficient for this CAI package. There is no restriction on selection of the keyboard and a standard typewriter format is sufficient.

The target population for this CAI package is high school (11-12 and selected), junior college students enrolled in a course in computer literacy. This package is menu-driven and easy to use. Students will have no trouble loading the CAI package with its self-explanatory capability.

Determining the Content of Instruction

Determination of the instructions used in design and development of a CAI software package is a demanding and exacting task. Several textbooks, magazines, conference papers, and the related literature on "the content of a computer literacy course" were reviewed to identify the essential instructions necessary in design and development of a well-written CAI package. After an inspection the following instructions were determined to cover the topic:

1. Computer terms
2. Computer concepts
3. Computer hardware systems
4. Computer software
5. Information systems
6. Computers and society
7. BASIC programming

The instructional topics were divided into two different parts, Computers Today and BASIC Programming. Computers Today, involved the first six topics, and facilitated student learning with numerous drills and practices. The BASIC programming utilized the tutorial method to teach the students the basic commands used in writing a BASIC computer program. Both part A and part B consists of six lessons as following:

Part A: Computers Today

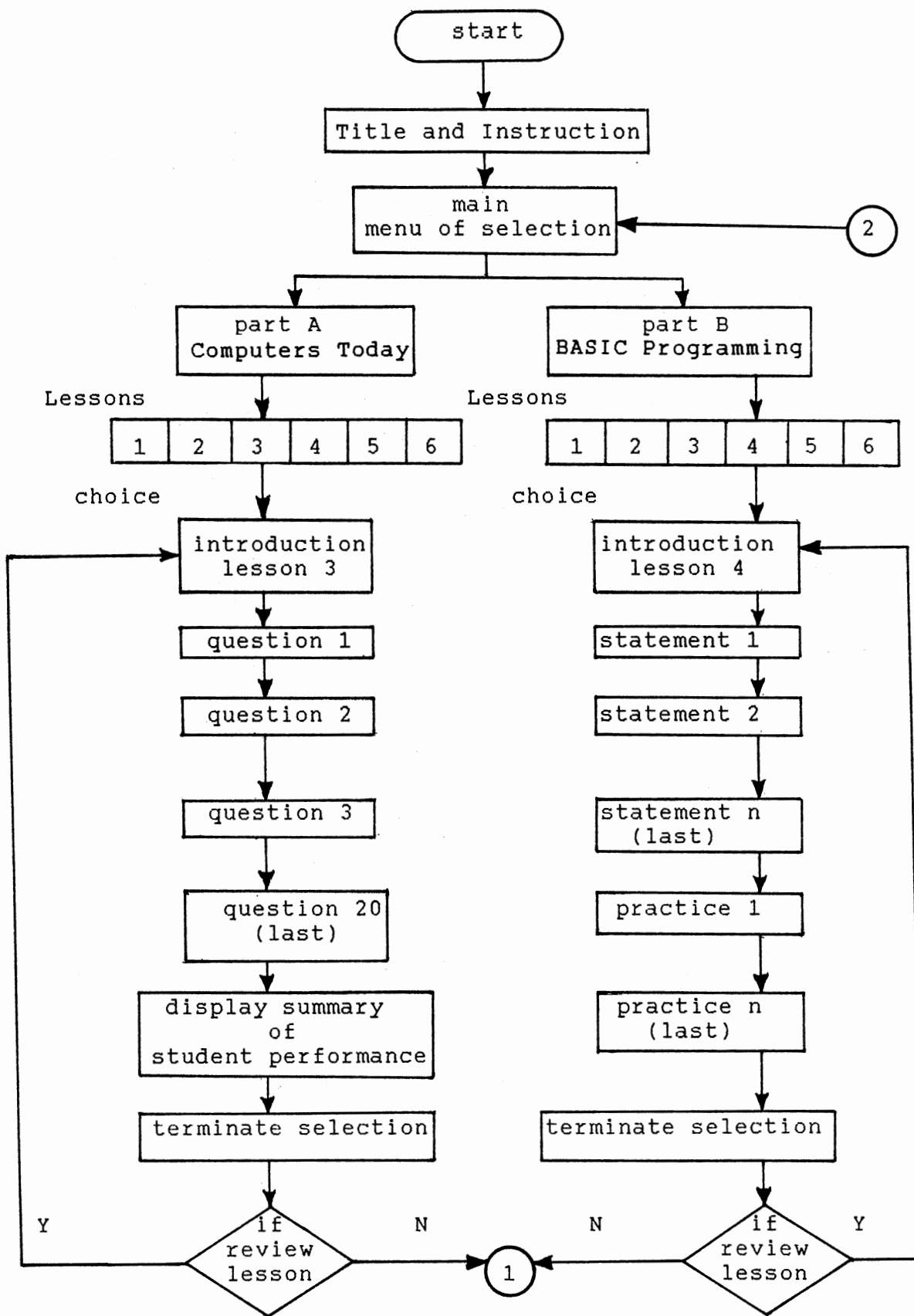
- lesson 1. Computer terms
2. Computer concepts
3. Computer hardware systems
4. Computer software
5. Information systems
6. Computers and society

Part B: BASIC Programming

- lesson 1. Getting started
2. Input and output
3. Control statement
4. Looping around
5. Subroutines and functions
6. Arrays

Designing and Laying Out the Instruction

The software developed in this study was designed to serve as a substitute teacher to guide the students and provide them with numerous problems related to computer literacy. The software organization can be best represented by the flow chart given in figure 3.



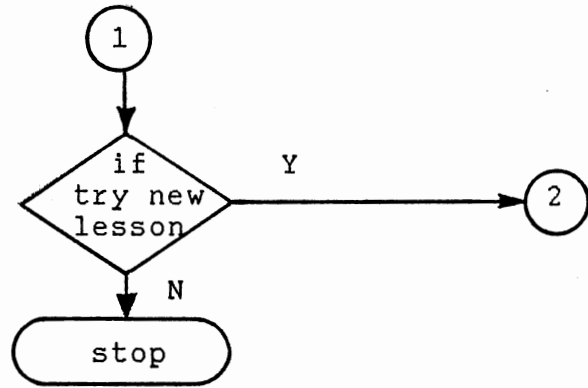


Figure 3. The Organization of the Software

This flow chart starts with a title to specify the content of the software. The title is followed by some preliminary instructions to guide students before the software is used. The software is menu-driven and provides a main menu consisting of two parts. Students can either choose part A, computers today, or part B, BASIC programming.

If part A is selected, the sub menu (lesson menu) in part A displays the listing of 6 different lessons. The student can select any of these lessons to improve his/her computer knowledge. These 6 lessons follow the same sequences, although their contents are different. Each lesson starts with an introductory section to provide complete and clear objectives and instructions for that lesson. Each lesson consists of 20 questions. The questions are multiple choice with four options and a value of 5 points each. Upon finishing the 20 questions, the student will receive a score and the number of questions answered correctly. Following the summary of student performance, the ending menu with three different options will be provided,

- (1) Review the same lesson
- (2) Try a new lesson
- (3) Exit the software

If the student selects option (1) the program will start at the introduction and proceed through the same lesson again. If option (2) is selected the program will start at the main menu. The third option will lead the student to the

closing remarks where the program is terminated.

Should part B be selected, the same sequence is followed. The only difference between instructions in part A and B is the presentation of tutorial lessons in part B instead of 20 questions. In addition, the tutorial lessons provide 5 sample problems for a learning check and student self-evaluation. This part also does not provide a summary of student performance because it is designed to tutor the students, not test them.

This software is designed to provide computer knowledge and some basic steps in BASIC programming for the students. Since the methods used in these two parts are completely different, the design procedures were divided into two distinct sections:

- (1) drill-and-practice design
- (2) tutorial design.

Drill-and-Practice Design

Drill-and-practice is used to facilitate the learning of material which requires additional reinforcement. The drill-and-practice design provides a better learning facility than classroom environment. It is used to help students maintain or strengthen knowledge they already have, such as spelling, symbols of chemical elements, and elementary arithmetic.

The drill-and-practice software is designed for students who are already enrolled in a computer literacy course. This

software covers six different lessons to help, test, and reinforce student understanding of each subject. The procedures of design and layout of instructions used in this software is based on the following steps,

1. Determining the number of questions
2. Determining the number of tries
3. Designing a system flow chart
4. Designing a detailed flow chart
5. Design and layout of frames

Determining the Number of Questions

It is important to select the size of lessons as long as it can be accomplished in a reasonable period of time without becoming discouraged. There is no standardized way of indicating the size of a lesson. The size of a lesson depends on the role of the questions in the instruction and the complexity of the material. If the concept is complex and covers several rules, such as the principle law of physics or the concepts of computer literacy, one question or one practice would not test the student's understanding adequately. In such cases, a set of questions need to be presented.

Statistics have shown (Alessi and Trollip, 1985) that a good lesson should be about ten to twenty minutes in length, with about sixty to seventy-five frames. In this thesis, all students are required to do the same fixed number of

exercises. This decision is most appropriate when each of the questions tests different knowledge, such as the memory unit in a computer. Knowing the functions of commands in read-only memory does not guarantee that students know the random access memory or the other storage units in a computer. Thus, the number of questions in each lesson were selected to be twenty.

Determining the number of tries

CAI designers often are concerned about how many times a student should be allowed to try to enter the correct response if the first one is incorrect. If the student is allowed only one attempt to answer the question, he or she may take the easy way and type anything just to satisfy the requirement of making a response, because the student knows that the correct response will appear shortly. Thus, it is reasonable to allow students more than one try. If the number of tries is more than two, the student probably just keyed in something as a guess. In this case, the student should be allowed to answer the question only twice, because it gives the student a chance to think what the correct answer is after the first attempt. As a result, the students are motivated to challenge and beat the machine.

The System Flow Chart

The system flow chart shows the major parts of the drill-

and-practice software used in design and development of the CAI software package for this study. This flow chart only shows "the big picture" and the details can be seen from the detailed flow chart. For example, a lengthy lesson made up of many frames (screen) can only be represented by a single block on the system flow chart, while the details of each frame are shown on the detail flow chart.

The system flow chart used in this study is shown in figure 4, page 60. This flow chart is arranged so that each lesson follows in an appropriate sequence. It can be seen that figure 4 is a part of the software organization drawn in figure 3. The first block in the flow chart, computers today, is dealing with the computer terms, concepts, software, hardware, information systems, and computers and society. The second block, lessons menu, provides the listing of six different lessons for the student selection. The third block, introduction, presents the objective and direction of each lesson. The detail of questions and summary of student performance is covered in the detail flow chart.

Detailed Flow Chart

The detailed flow chart is much more complete than the system flow chart. The lesson design can be shown in detail using the detailed flow chart. If the content or the instructional logic of the CAI software is very complicated,

the detailed flow chart may be needed to provide the detail of the lesson, otherwise the system flow chart should be sufficient.

A detailed flow chart can be created by adding details to the system flow chart. For example, the detailed flow chart in figure 5, pages 61-62, shows the whole sequence of instructions used for each question, while these sequences are shown by a block in the system flow chart.

The detailed flow chart in figure 5 starts with the title and instructions of the lesson. The number of question, Q , and the number of tries, T , are initially set to zero. Each question is displayed and the student is allowed to respond twice if the first response is incorrect. If the correct response is entered, the student will receive a reward; otherwise the appropriate feedback is given based on the student selection. If the student presses any unexpected keys the program will not abort and the response is ignored. In this case, students are allowed to enter their response again.

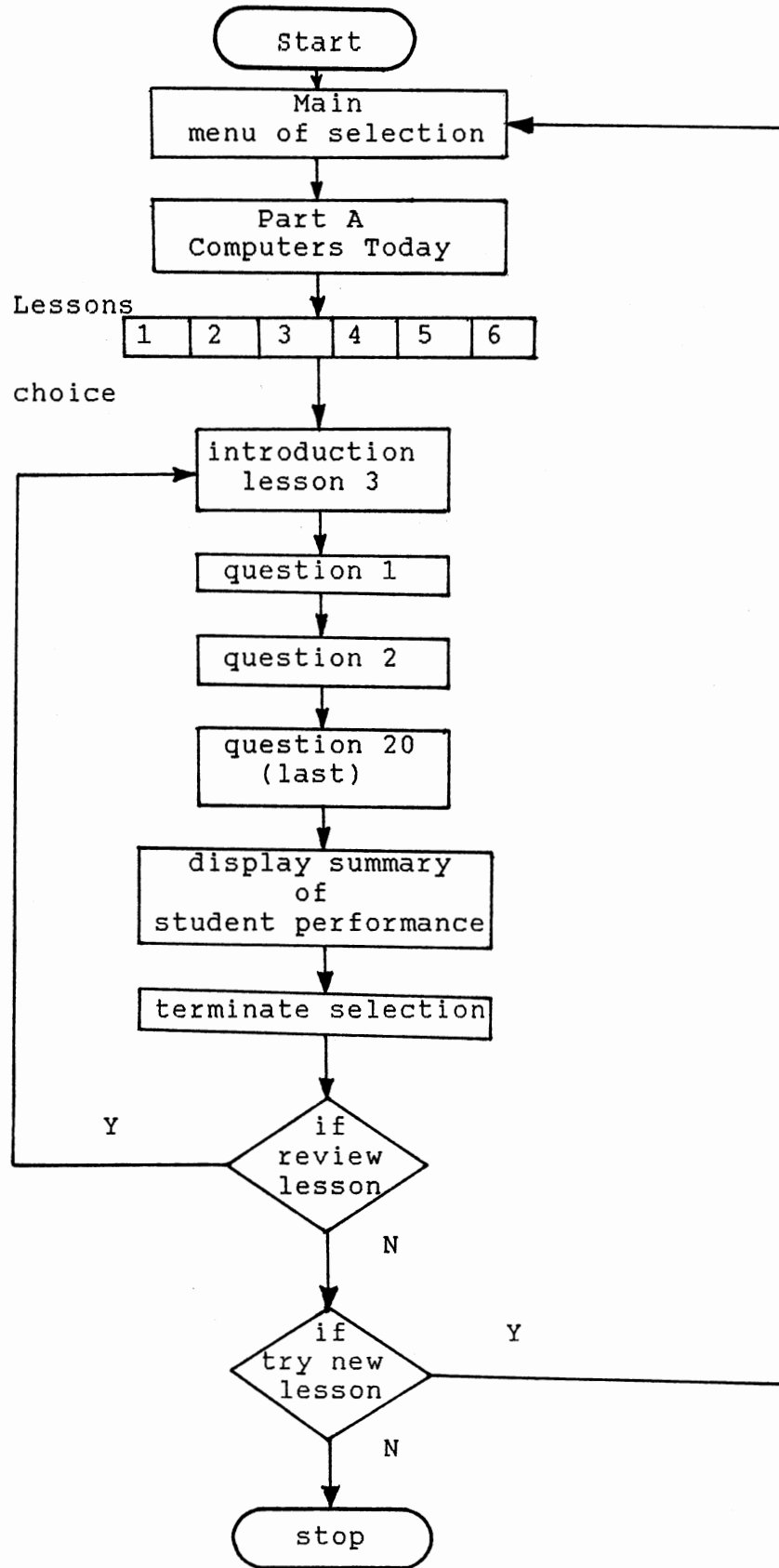
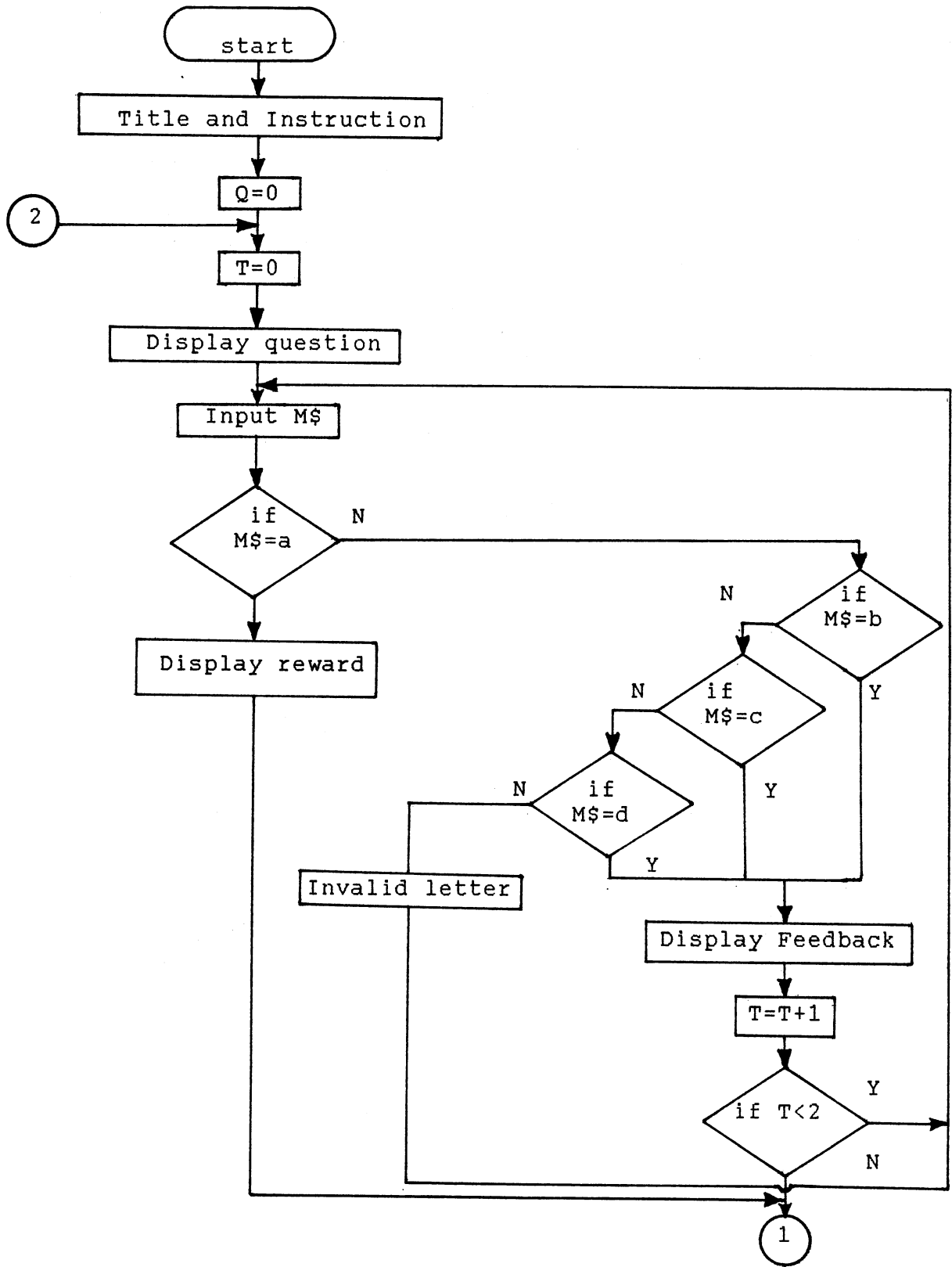


Figure 4. System Flow Chart of Drill-and-Practice



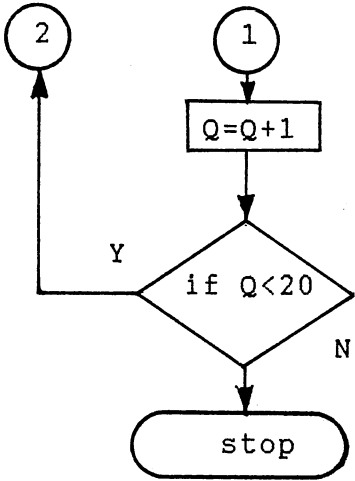


Figure 5. Detailed Flow Chart of Drill-and-Practice

Design and Layout of Frames

The frame is referred as a single page of the instructional program. In CAI software, the frame is defined as one screen of the program. A frame may contain a question, instructions, score, or the test procedures. The concept of frames is often used as one unit of measure in CAI lessons. The major factor of successful instructional software is well-designed frame types. Frames used in this study are classified into seven different categories.

1. Lesson-menu frame
2. Introductory frame
3. Question frames
4. Entry-error frame
5. Score frame
6. Ending-menu frame

Lesson-Menu Frame. The lesson-menu frame in figure 6, page 64, provides the listing of the selected lessons for student choice. The students are allowed to select any lesson. The displayed menu provides an opportunity for the student to select a lesson that meets his or her background. In addition, computers have better memories than people and always remember things exactly as they were stored. A good program should rely on computer memory as much as possible. Each lesson is selected from a displayed menu. This option relies on the proper usage of computer memory and minimizes the student memory demands.

Part A

COMPUTERS TODAY

MENU

- <1>. COMPUTER TERMS
 - <2>. COMPUTER CONCEPTS
 - <3>. COMPUTER HARDWARE SYSTEMS
 - <4>. COMPUTER SOFTWARE
 - <5>. INFORMATION SYSTEMS
 - <6>. COMPUTERS AND SOCIETY
- YOUR CHOICE (1-6) ?

Figure 6. Lesson-Menu Frame of
Drill-and Practice

Introductory Frame. The introductory frames appear at the beginning of each lesson. Samples of these frames are shown in figures 7 and 8, pages 65-66. They introduce the objective of the lesson and direct the student how to move forward. They provide the purpose and contents of the lesson. Directions are essential to the effective use of any CAI software. They should include information about how to answer questions and what the important instructions are. The introductory lesson also collects some information from the student, such as name, which can be used later in the program to personalize the instruction.

```
*****  
*  
*   WELCOME TO YIN-HWA SOFTWARE COMPANY   *  
*  
*****
```

This software package is designed for a course on computer literacy. It has been divided into two parts,

- (A) Computers today
- (B) BASIC programming

Part (A) consists of six lessons with 120 drill problems.

Part (B) also consists of six lessons. This part is designed to be a tutorial lesson for BASIC programming. Each lesson provides 5 problems for a learning check.

ENTER YOUR NAME TO BEGIN ?

Figure 7. Introductory Frame of CAI Software

Computers have become highly visible tools in our modern society. People relate to these tools with an extraordinary range of views and emotions.

The multiple-choice questions which follow will help, test, and reinforce your understanding of the subject: COMPUTERS AND SOCIETY.

You will be given 20 questions, each worth 5 points. After each question, you will see a "?" prompt. Type your answer, then press the RETURN key.

You are allowed two attempts for each question. If your first answer is incorrect, please wait and do not press any key. The screen will be cleared and the same question will reappear for the second time.

Upon finishing the 20 questions, you will be given the number you answered correctly and your score.

GOOD LUCK ! <user name>

PRESS ANY KEY TO CONTINUE

Figure 8. Introductory Frame of Drill-and-Practice

Question Frames. The question frame is a common type frame used in this instruction. This frame clearly shows all sequences that are performed to answer a multiple choice question by the student. These sequences occur in the following consecutive order:

1. display of a question
2. prompt for the answer
3. student response
4. judgment of response
5. feedback.

First, a question is displayed. Then the student enters a response after the prompt "?." The response must be entered in lower case. The computer checks the student response and provides feedback to the student. If the response is correct, the feedback will be a short affirmation followed by pleasant music. Otherwise, the proper feedback based on the student response is given. The student is allowed to try once again. If the response is still incorrect, the correct answer will appear on the screen. Figures 9-11 show the correct response after the first try, second try, and incorrect response, respectively.

The feedback for the incorrect response should be corrective, positive, reinforcing, informative, diagnostic, and prescriptive. If the feedback is related to the student response, it will produce the proper explanations based upon the individual response. Thus, the student can be lead to understand why the response was incorrect.

1. Which computer language consists of strings of binary numbers and is the only one the CPU directly understands?

- a. COBOL
- b. FORTRAN
- c. Machine
- d. none of the above

? c

THAT'S GREAT! <user name>,
YOU ARE OFF TO AN EXCELLENT START!

PRESS ANY KEY TO CONTINUE

Figure 9. Correct Response at the First Try

1. Which computer language consists of strings of binary numbers and is the only one the CPU directly understands?
- a. COBOL
 - b. FORTRAN
 - c. Machine
 - d. none of the above

? a

SORRY, INCORRECT !!!

<user name>, COBOL is designed specifically for business-type data processing applications.

PLEASE TRY AGAIN

<timer setting>

1. Which computer language consists of strings of binary numbers and is the only one the CPU directly understands?
- a. COBOL
 - b. FORTRAN
 - c. Machine
 - d. none of the above

? c

THAT'S GREAT! <user name>,
YOU ARE OFF TO AN EXCELLENT START!

PRESS ANY KEY TO CONTINUE

Figure 10. Correct Response at the Second Try

1. Which computer language consists of strings of binary numbers and is the only one the CPU directly understands?
- a. COBOL
 - b. FORTRAN
 - c. Machine
 - d. none of the above
- ? b

SORRY, INCORRECT !!!

<user name>, FORTRAN is designed for scientific applications.

PLEASE TRY AGAIN.

<timer setting>

1. Which computer language consists of strings of binary numbers and is the only one the CPU directly understands?
- a. COBOL
 - b. FORTRAN
 - c. Machine
 - d. none of the above
- ? d

SORRY, INCORRECT !!!

The answer is:

(c) Machine

PRESS ANY KEY TO CONTINUE

Figure 11. Incorrect Response for Both Tries

Entry-Error Frame. Good instructional software should take into account the possibility of all responses entered by the students. If an unexpected key is pressed, the program should identify the error and direct the student how to enter the response. For example, if the program requires the lower case letter a, b, c, and d for a multiple-choice question, the lower case letters from e-z and all upper-case letters are considered to be unexpected characters. Should the student enter an unexpected character, the computer would inform him or her to enter the right character. The entry-error frame shown in figure 12, page 72, demonstrates the ability of the computer to recognize unexpected characters and method of directing the student to enter the response.

Score Frame. Once the student finishes all the questions in each lesson, the computer will check the responses and calculate the student score. The score frame in figure 13, page 73, shows the total score to stimulate the student and to encourage him or her for improvement. In addition, the displayed score will motivate the student to be competitive with himself or herself and provide a self-evaluation function.

Ending-Menu Frame. The student should be able to terminate the drill section at any time. A simple procedure is available for doing this. The direction for termination, pressing the Ctrl and Break keys simultaneously, is given at the beginning of the introductory section. If the students

do not remember how to terminate a drill correctly, they will probably turn off the computer, remove the diskette, or just walk away. In this study, the students are informed how to terminate at the beginning and the end of each lesson.

When a lesson is finished, a good program should provide the options of reviewing the same lesson, trying another one, or exiting. The ending-menu frame shown in figure 14, page 74, provides all three options. Those students who have performed their lesson very well don't need to review the same lesson again. They are allowed to try another lesson or to exit. But those who performed poorly might need to review the same lesson. Some students may get frustrated and want to exit at the end of the first lesson. Once the exit option is selected, a short final message will appear on the screen (see figure 15, page 74). Having these three options is reasonable and increases the flexibility of the material.

1. Which computer language consists of strings of binary numbers and is the only one the CPU directly understands?
 - a. COBOL
 - b. FORTRAN
 - c. Machine
 - d. none of the above

? t

INVALID LETTER, PLEASE ENTER a, b, c, or d

Figure 12. Entry-Error Frame of Drill-and-Practice

<user name>, YOU HAVE ANSWERED 18 QUESTIONS CORRECTLY OUT OF
20 QUESTIONS!! YOU HAD 90% CORRECT ANSWERS!

If you are unsure of some of the questions, please ask
your instructor to explain them.

If your score was less than 80%, you should review this
lesson !!

PRESS ANY KEY TO CONTINUE

Figure 13. Score Frame of Drill-and-Practice

WHAT'S NEXT ?

You have just completed this lesson. You now have three options:

1. Review this lesson
2. Try a new lesson
3. Exit

(select 1-3)

?

Figure 14. Ending-Menu Frame of Drill-and Practice

THANK YOU FOR YOUR ATTENTION.

GOOD BYE! HAVE A NICE DAY !

Figure 15. Closing Frame of Drill-and Practice

Tutorial Design

The software design and programming techniques of tutorials are more complicated than drill-and-practice, but the logic of tutorial software is much simpler than drill-and-practice. The complex part of the tutorial software design is the presentation of the material and the method of guiding the student. The heart of tutorial software is a variety of written explanations, descriptions, examples, and graphic illustrations for the concept development rather than a variety of questions used for the drill-and-practice.

This software covers six different lessons to teach the student the general statements of the BASIC language. The procedures of designing and laying out of instructions used in this software are based on the following 2 steps,

1. Designing the system flow chart
2. Design and layout of frames.

Designing the System Flow Chart

The instructional logic of the tutorial software is not as complicated as the drill-and-practice software. In this case, the detailed flow chart is not necessary and the system flow chart will provide sufficient information.

The system flow chart in figure 16, page 77, starts with the main menu where the student selects BASIC programming. Then the computer provides the lesson menu which displays the listings of six different lessons. The students can select

any of these lessons to improve their programming ability. Each lesson starts with an introductory section to provide a complete and clear objective of the lesson. Each lesson consists of several statements which are used in BASIC programming. These statements are generally taught using step-by-step descriptions and demonstrations through several teaching frames. Once all statements are taught, five multiple-choice questions are provided to check student progress.

First, the number of questions is initially set to zero. Then, each question is displayed and the student is allowed to respond once. If the correct response is entered, the student will receive a reward; otherwise the appropriate feedback is given based on the student selection; and the correct answer will appear on the same frame. Upon finishing the five questions, the ending menu will be provided with three different options: review the same lesson, try a new lesson, or exit the software.

Design and Layout of Frames

The tutorial software has the lesson-menu frame, introductory, question, entry-error, ending-menu and closing frames. The principles and procedures used in design and development of these frames are similar to those discussed in Drill-and-Practice software design. The detail of design and layout of the tutorial frames are shown in figures 17-21, pages 78-82.

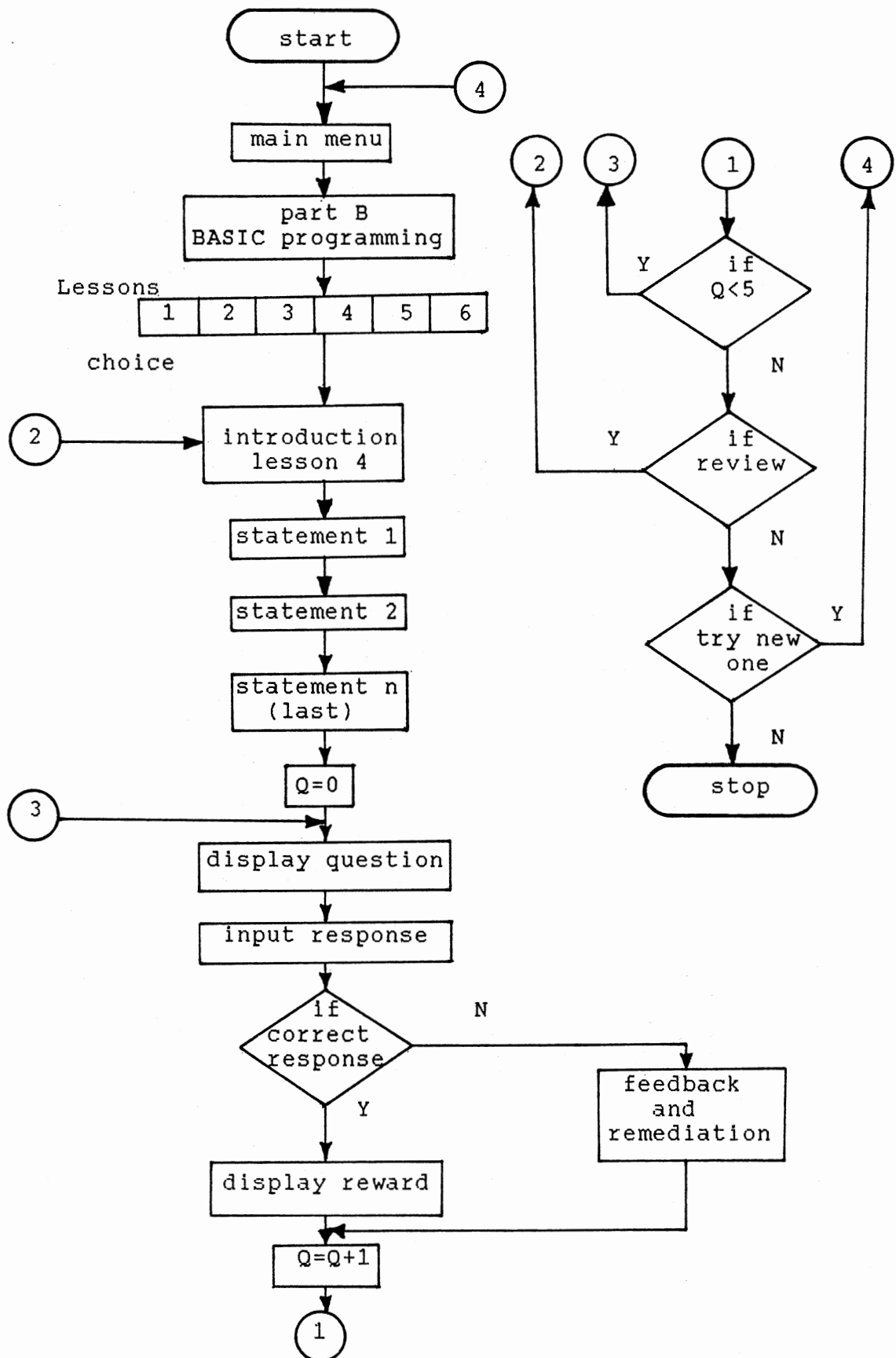


Figure 16. The System Flow Chart of Tutorial

Part B

BASIC PROGRAMMING

MENU

<1>. GETTING STARTED

<2>. INPUT AND OUTPUT

<3>. CONTROL STATEMENT

<4>. LOOPING AROUND

<5>. SUBROUTINES AND FUNCTIONS

<6>. ARRAYS

YOUR CHOICE (1-6) ?

Figure 17. Lesson-Menu Frame of Tutorial

There are several ways to store information in address locations. In lesson 1, you have learned to assign the information by using the following LET statement:

```
LET A=4    or    LET A$="COMPUTER."
```

In this lesson, you will learn two more ways to store information.

First, use of an INPUT statement, which allows the user to key data into the computer to continue running the program.

Second, use of the READ/DATA statement allows previously stored data to be read.

Also, you will learn how to use the PRINT function to produce an organized and formatted output with headings, columns and appropriate spacing.

PRESS ANY KEY TO CONTINUE

Figure 18. Introductory Frame of Tutorial

WHAT'S NEXT ?

You have just completed this lesson. You now have three options:

1. Review this lesson
2. Try a new lesson
3. Exit

(select 1-3) ? 4

INVALID LETTER, PLEASE ENTER AGAIN.

Figure 19. Entry-Error Frame of Tutorial

3. What is the output of the following program?

```
10. READ A,B,C$,D
20. READ T$,S,Y
30. DATA 36,40,NEW
40. DATA 42,ITEM
50. PRINT S
60. END
RUN
```

- a. 40
- b. OUT OF DATA
- c. ITEM
- d. 36

? b

PRETTY GOOD ! <user name>, I KNEW YOU COULD DO IT!

PRESS ANY KEY TO CONTINUE

Figure 20. Correct-Response Frame of Tutorial

3. What is the output of the following program?

```
10. READ A,B,C$,D
20. READ T$,S,Y
30. DATA 36,40,NEW
40. DATA 42,ITEM
50. PRINT S
60. END
RUN
```

- a. 40
- b. OUT OF DATA
- c. ITEM
- d. 36

? d

SORRY, INCORRECT.

The correct answer is "b"

If there is not enough data in the DATA list, an OUT OF DATA error message will appear and the program stops executing.

PRESS ANY KEY TO CONTINUE

Figure 21. Incorrect-Response Frame of Tutorial

In addition, the tutorial software has many teaching frames that drill-and-practice does not have. These frames are used to describe the statements of the BASIC programming language and can be classified into three different categories based on the method of presentation.

RULEG Frames. Rules or principles play an important role in the mathematical, physical, and social sciences. Students can learn a rule or a principle from a definition alone, but they usually need examples to help them fully understand it (Steinbery, 1984). Examples should illustrate the full range of the concepts, analyze its critical and its variable attributes, and add to the student's understanding. RULEG is an acronym in which the instructional logic proceeds from the rule (RUL) to example (EG). It is referred to as a deductive process. The rules or principles and their foundations are directly stated and then demonstrated by the proper examples.

In the RULEG method, the students are taught the laws and principles first and then asked to apply these laws, principles, or any other form of rules to a set of examples. The teaching frames shown in figure 22-23, pages 85-86, can best illustrate the RULEG design.

EGRUL Frame. If the instructional logic proceeds from example (EG) to rule (RUL), the teaching approach is referred to as EGRUL. This approach demonstrates applications and leads the student to discover the rule or principle. Typically, the students would be provided with some examples to determine the rule or principle of some phenomena. The

EGRUL is a form of discovery learning obtained from a set of examples. It is referred to as an inductive process. The teaching frames shown in figures 24-27, pages 87-90, are used to illustrate the EGRUL method.

Comparison Frames. Once the rules, principles, and fundamentals of the BASIC statements have been taught, the comparison method is used to improve the student programming ability. In this approach, two programs are compared which have different source codes but produce the same output. This method provides the art of programming technique and motivates the students to achieve the skill which is required for writing efficient source code. The teaching frames displayed in figures 28-29, pages 91-92, show two different programs but have the same output.

Creating Graphics

Graphics plays an important role in the learning process. Instructional software is less interesting than other software and can quickly become boring. If a student is not performing well, it can become frustrating. Graphics and color provide one way to increase motivation.

In this study, each title page is decorated with some graphics related to the selected lessons to interest the student and to prevent boredom. Once the program is loaded, the title frame with graphics will be displayed. After a few seconds the screen will be cleared and the introductory frame will appear. In addition, the graphics are used in tutorial lessons to help the students understand the statement and

the logic of programming. For example, the flow chart is the best graphics representation that is used to explain the major parts of the software.

The medium and high resolution graphics facilities are used to produce the graphics utilized in CAI software design. This graphics should be designed and layout on a paper before coding. Special graphic sheets for the layout can make the graphics products more efficient and reduce the debugging time. Two title frames decorated with the proper graphics are shown in Figures 30-31.

IF/THEN/ELSE

The general format of the IF/THEN/ELSE statement is:

```
Line# If condition THEN clause ELSE clause
                (line#)         (line#)
```

For example:

```
10 IF GPA > 3.5 THEN 50 ELSE 60
50 PRINT "GOOD"
60 PRINT "NEEDS IMPROVEMENT!"
```

or

```
10 IF GPA > 3.5 THEN PRINT "GOOD" ELSE PRINT "NEEDS
IMPROVEMENT!"
```

A sample program with IF/THEN/ELSE statement and its flowchart are given in the next screen.

PRESS ANY KEY TO CONTINUE

Figure 22. RULEG Frame 1 of Tutorial

ARRAY

A set of subscripted variables is called an ARRAY.

For example, there are five students whose test scores are 98, 72, 88, 64, 93. The scores could be put in an ARRAY called A:

98	72	88	64	93
----	----	----	----	----

A(1) A(2) A(3) A(4) A(5)

Also can be read:

A(1)=98

A(2)=72

A(3)=88

A(4)=64

A(5)=93

Each single value (containing the score value) is called an element.

PRESS ANY KEY TO CONTINUE

Figure 23. RULEG Frame 2 of Tutorial

```
WHAT'S YOUR NAME ?  
? Linda
```

```
HELLO, Linda
```

Do you want to know how the computer accepts your response?

Look at this program:

```
10 PRINT "WHAT'S YOUR NAME ?"  
20 INPUT T$  
30 PRINT "HELLO, "; T$  
40 END  
RUN  
WHAT'S YOUR NAME?  
? Linda  
HELLO, Linda
```

-----> INPUT tells the computer to
display a question mark
and wait for user's response
(assign the string to the T\$)

-----> User's response.
The semicolon continues the output
on the same line.

PRESS ANY KEY TO CONTINUE

Figure 24. EGRUL Frame 1 of Tutorial

INPUT

Causes the computer to request data from the keyboard.
This is another way to assign the value to the address location.

Here is another program which uses INPUT statements. Look at the program and the output.

```
10 PRINT "TYPE TWO NUMBERS:";
20 INPUT A,B
30 PRINT " THE NUMBERS ARE:";A,B
40 PRINT "     THEIR SUM IS:";A+B
50 PRINT "THEIR PRODUCT IS:";A*B
60 END
RUN
TYPE TWO NUMBERS: ?15,17
  THE NUMBERS ARE: 15           17
    THEIR SUM IS: 32
  THEIR PRODUCT IS: 255
```

Line 20 INPUT A,B will call for two numbers which are typed by the user, separated by a comma.

PRESS ANY KEY TO CONTINUE

Figure 25. EGRUL Frame 2 of Tutorial

What is the capital of Arizona?
a. Tucson
b. Tempe
c. Phoenix
d. Austin

? a
SORRY, INCORRECT!

How did the computer decide whether the user's answer was right?
The following program and output will show you how it works.

```
10 PRINT "What is the capital of Arizona?"
20 PRINT "a. Tucson"
30 PRINT "b. Tempe"
40 PRINT "c. Phoenix"
50 PRINT "d. Austin"
20 INPUT M$
30 IF M$="c" THEN PRINT "YOU GOT IT"
40 IF M$<>"c" THEN PRINT "SORRY, INCORRECT!"
50 END
```

```
RUN
What is the capital of Arizona?
a
SORRY, INCORRECT!
```

PRESS ANY KEY TO CONTINUE

Figure 26. EGRUL Frame 3 of Tutorial

IF/THEN

The general format of IF/THEN statement is:

Line# IF condition THEN line#
(statement)

For example:

100 IF A=2 THEN 300

The following are relational symbols that can be used:

SYMBOL	CONDITION	EXAMPLE
<	less than	A < B
>	Greater than	A > B
=	Equal to	A = B
<=	Less than or equal to	A <= B
>=	Greater than or equal to	A >= B
<>	Not equal to	A <> B

PRESS ANY KEY TO CONTINUE

Figure 27. EGRUL Frame 4 of Tutorial

Compare the following two programs:

A.

```
10 LET A=1
20 PRINT A;
30 IF A=6 THEN 100
40 A=A+1
50 GOTO 20
60 END
RUN
1 2 3 4 5 6
```

B.

```
10 FOR A = 1 TO 6
20 PRINT A;
30 NEXT A
40 END
RUN
1 2 3 4 5 6
```

Program B only uses 4 statements to produce the same output as program A, because Program B uses a FOR/NEXT statement instead of a control statement.

PRESS ANY KEY TO CONTINUE

Figure 28. Comparison Frame 1 of Tutorial

Let's compare these two programs:

A.

```
10 PRINT "What is the capital of Texas?"
60 INPUT MS
70 IF MS="AUSTIN" THEN PRINT "YOU GOT IT"
80 IF MS<>"AUSTIN" THEN PRINT "SORRY, INCORRECT!"
90 END
```

```
RUN
What is the capital of Texas?
AUSTIN
YOU GOT IT!
```

B.

```
10 PRINT "What is the capital of Texas?"
20 INPUT MS
30 IF MS="AUSTIN" THEN 40 ELSE 50
40 PRINT "YOU GOT IT!" : GOTO 60
50 PRINT "SORRY, INCORRECT!"
60 END
```

```
RUN
What is the capital of Texas?
AUSTIN
YOU GOT IT!
```

The two programs outputs are the same.
But, line 30 of program B has a new statement IF/THEN/ELSE.

PRESS ANY KEY TO CONTINUE

Figure 29. Comparison Frame 2 of Tutorial

EXERCISES U

INFORMATION SYSTEM

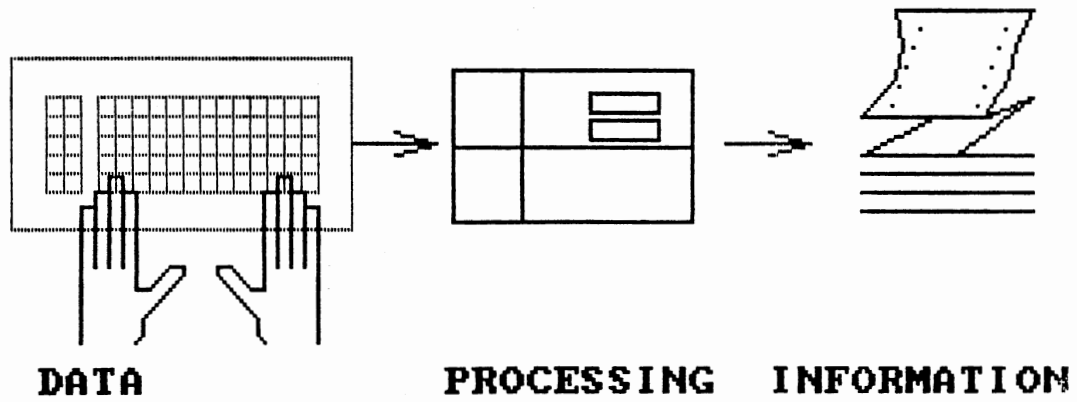


Figure 30. Graphics Frame 1 of CAI Software

SUBROUTINE #####
SUBROUTINE #####
SUBROUTINE

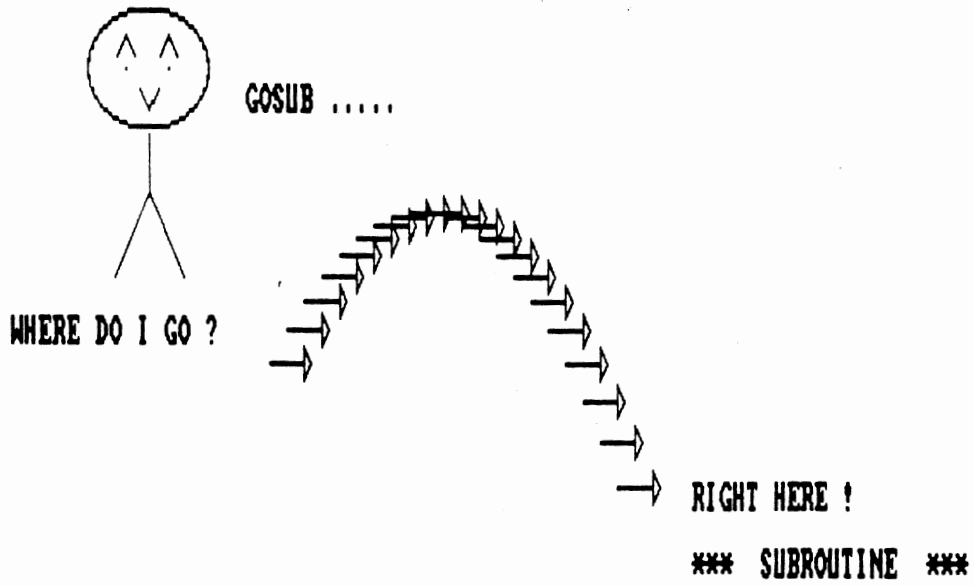


Figure 31. Graphics Frame 2 of CAI Software

Coding, Testing, and Debugging

Coding is the process of writing the programming instructions. It is a time-consuming process and is used to submit programs to the computer. The method of coding is highly dependent on the system being used. It is a more or less difficult process depending on the language chosen. In any case, the process is essentially the same.

The overall consideration in coding is always to produce a high-quality instructional lesson. The efficiency of the programming and efficiency of execution of instructions should never be chosen over the quality of the lesson. In addition, it is beneficial to make the lesson function properly before trying to optimize the programming code.

The program code can be prepared either on-line or off-line. In off-line, a substantial part of the program is written on paper before putting it into the computer. But in on-line code, the program is written directly into the computer without using any paper. The CAI software program utilizes both on-line and off-line facilities. The majority of coding in this study has been done through an on-line process. This process benefits the frames design to speed up the coding. Using the frame design, each segment of the program is typed directly into the computer from the keyboard and has been tested immediately.

Once the program has been written, it must be checked for errors. This testing is often referred to as debugging. Debugging is the process of locating and correcting program

errors. In this study all independent segments of code were tested as soon as they had been programmed. For example as soon as a subroutine was programmed, it was tested to see that it functioned as desired.

The common errors appearing on programs can fall into two different categories: (1) syntax errors and (2) logic errors. The syntax errors are usually typing mistakes made during the coding process. This error can be removed or greatly reduced by carefully proofreading program statements as they are typed. Once the program is free of any syntax error, it is able to run on the computer. This program needs to be tested with a variety of data to determine if the results are always correct. If a program obtains correct results with one set, but incorrect results when run with a different set of data, the type of error is referred to as a logic error.

In this study, the software were divided into thirteen files to reduce the memory requirement. The program was finished through a continuing debugging process and the files were stored on a floppy diskette. The software was then ready to run on the ADVANCED BASIC mode for demonstration.

Compiling and Linking

The CAI software package designed for this study is written in the BASIC programming language. This language is fairly easy for people to understand, but the CPU cannot interpret this language. The CPU, which executed the

instructions, can only understand machine language. Therefore, a program written in BASIC must be translated into machine language before it can be executed on the CPU. The program is translated into machine language by a system program called a compiler. The compiler creates an object program, which consists of an entire program translated into machine language (Mandell, 1987). The object program can be loaded into the computer's memory for execution after the link process.

The link program is used to create an executable program using a collection of the object files. The executable program is a file which can be run on the CPU. Once the process of compiling and linking is completed, the CAI software is ready for student use.

The following instructions should be followed to run the CAI software package developed in this study:

1. Place the DOS diskette in drive A.
2. Turn on the machine and wait until the system is booted up.
3. Enter the current date and time. The A prompt, A > , will be displayed.
4. Replace the CAI diskette in drive A and type CAI. The program will be loaded and the title page will be displayed with a pleasure music. The user need not do anything but follow the instructions.

CHAPTER V

SUMMARY AND RECOMMENDATIONS

Summary

The advanced technology and the introduction of the microcomputer have greatly influenced the U.S. educational system. The low cost and powerful microcomputers provide an opportunity for teachers and students to be computer literate and to teach them the fundamental concepts of computer operations and programming.

The computer technology has played a major role in information processing and data communications. The recognition and understanding of the important features of computers in an information society requires an educational effort to remove or greatly reduce computer illiteracy. The term "computer literacy" is generally referred to as the required knowledge, skill, and attitudes that one needs to have to function efficiently and effectively in the performance of computer related operations. Students need to know about the capabilities and the limitations of computers. The recognition and understanding of various features of the computers help them to realize that computers do not have any intelligence and they only perform the tasks they are

instructed to do.

The primary goal of computer usage in education is for the delivery of instructions. The learner and computer are in direct communication with each other. This process is called the "interaction," which can be obtained by a well-written CAI program. The CAI program is capable of modifying the presentation of the instruction to match the capabilities of the individual learner, while the other forms of instruction do not involve interaction and proceed at a fixed pace, with no consideration for the needs of individual learners. In addition, the advantage of CAI over the traditional method of instruction is the individualization. Each student can control the pace of learning, which is difficult in the lecture situation. The CAI software is designed to make learning an active process, with students playing a constant thinking role, which is superior to a passive learning process.

There are a number of ways that the computer can be used in the instructional process. The primary usage of computers in each application solely depends upon the nature of the work. An elementary student practicing a vocabulary game on the computer or an astronaut performing the landing on the surface of moon through the use of computer simulation are directly involved in CAI. The requirements of using the computer as a vehicle and the needs of computer-interaction are clearly stated in these cases.

Computer-assisted instruction will benefit both teachers and students when correctly implemented. Unfortunately, at the present time, there is a critical shortage of software for CAI. The rapid spread of microcomputers, which have several advantages over mainframe computers, has created a tremendous demand for CAI. Educators often can't acquire CAI software that is educationally suitable, relevant to their school's needs, and available at a reasonable price. In addition commercial software has a lack of instructional flexibility and even minor modification is not possible. These limitations and many other obstacles created a major crisis for CAI which can be resolved by teacher-made software.

The result of this is a critical need to create CAI software to run on microcomputers. This study covers the design and development of a CAI software package to provide an efficient and effective instructional program for a course on computer literacy.

The CAI software package developed in this study can be used as a assistant teacher for a course on computer literacy. This package consists of integrated computer literacy concepts and computer programming with the BASIC language. It can be used as a useful guidance tool for students who study computer literacy. It also should be useful to the individual teacher, student, industrial trainer, or computer professional who intends to develop an

instructional computer program. The CAI package is designed to satisfy the following objectives

1. Presenting information
2. Guiding the student
3. Provide the chance for students to practice
4. Assessing student learning which are used to assist teachers and to motivate students.

The related research papers and literature were reviewed to complement the background information related to CAI software and to determine if a similar study has been done. This study has emphasized the following topics:

1. The history of computers in education
2. The characteristics of Computer-Assisted Instruction
3. The factors of success in design and development of Computer-Assisted Instruction
4. The current status of computers in education
5. Trends of Computer-Assisted Instruction.

In order to satisfy the purpose and objectives of this study, numerous drill problems related to basic knowledge about the computer have been designed. In addition, the tutorial lessons are used to guide the students and to present information related to BASIC programming.

This package is designed to run on the IBM PC and IBM PC compatible with a Dos 2.1 or later version. An IBM PC or compatibles with graphics adapter, single disk drive, and minimum of 64 KB of program memory is sufficient for this CAI

package.

The proposed CAI software package is divided into two different parts, Computers Today and BASIC Programming. Computers Today facilitates student learning with numerous drills and practices. The BASIC Programming utilizes the tutorial method to teach the students the basic commands used in writing a BASIC computer program. The results of selected lessons of this study are shown in Appendix A and B. This software is stored on a single, double side/double density, 5 1/4 inch diskette with an executable format. The software is menu driven and a clear set of instructions for the users is provided.

A new software package for computer-assisted instruction has been developed. This software uses various methods of stimulations, sounds, graphics, and colors, to remove or greatly reduce computer illiteracy. The advantages of the software over traditional instruction includes:

1. Providing the optimal way to learn

Through using this software, the teacher can provide appropriate educational opportunities to all students, including both the academically talented and those who have difficulties in the learning process.

2. Providing individualized instruction

This software package is menu-driven which provides the main-menu, lesson-menu, and ending-menu for the learning control. It allows the student to control the learning

process and to work at his or her pace. It also improves achievement in terms of increasing motivation, decreasing anxiety, and improving student attitude.

3. Providing individualized interaction

This software package provides a variety of feedback related to student response followed by proper explanations based on the individual response. Students who use this software can easily communicate with the computer through the interactive process and move the progress.

4. Facilitating an infinite practice

One of the most important features of the CAI software designed for this study is the option to review the lessons as many times as desired. This feature makes the CAI software superior to the traditional method of instruction.

5. Increasing the motivation to learn

The CAI software package designed for this study utilizes graphics, color, and sound to increase motivation and to make the instruction more enjoyable. For example, the pleasant music indicating the beginning of the LOOP lesson can increase student curiosity and increase motivation.

6. Providing the instructional software for a large number of students

Through time sharing, a large number of students can access a computer simultaneously and work on the same software package. Each student can proceed at his or her own speed at the same time.

Recommendations for Future Research

Several possibilities exist for expanding the CAI software package developed in this study. The following recommendations are based on the results of the instructional software package on computer literacy and on the review of the related literature.

1. A similar study should be made to provide an effective instructional software package for teachers of computer literacy.
2. Lesson validation of this study, which is the process of checking how well the lesson works in an actual instructional setting, is needed to obtain data that will indicate which parts of the software need to be improved.
3. Studies of voice-input are needed to provide a voice-feedback for student responses. Human speech is the voice-input while the voice-feedback is the computer generated sound used for an advanced CAI. The result of this modification would greatly stimulate student motivation.
4. In-depth studies of storing the student's record should be made to maintain a record of student responses. These studies are used to keep track of the learner's progress.
5. The conversion of the CAI software into an instructional software utilized by Apple personal

computers is recommended. This software would provide an opportunity for the Apple users to be familiar with the CAI and to prevent additional cost for high schools that are not equipped with IBM PC or IBM PC compatible computers.

6. The various methods of stimulation for BASIC programming need to be studied. These methods can provide an environment where the student learns by performing the activities necessary to be learned in the real world.

4/24/80

BIBLIOGRAPHY

- Alessi, Stephen M., and Trollip, Stanley R., Computer-Based Instruction: Methods and Development, Prentice-Hall, Inc., 1985.
- Anderson, R. E. & Klassen, D. L. A Conceptual Framework for Developing Computer Literacy Instruction. AEDS Journal, 1981, 16(2), 128-150.
- Atkinson, Richard C. "Futures: Where will Computer-Assisted Instruction (CAI) Be in 1990?", Educational Technology, Vol.18, No.4, April 1978. pp.60
- Attala, Emile E. "Futures: Where will Computer-Assisted Instruction (CAI) Be in 1990?", Educational Technology, Vol.18, No.4, April 1978. pp.61
- Backer, Henry Jay. "Our National Report Card: Preliminary Results form thr New John Hopkins Survey", Classroom Computer Learning, January 1986. pp.30
- Barrette, Pierre. Microcomputers in K-12 Education Conference Proceedings. Computer Science Press, 1982.
- Beekman, George. "The Microcomputer as Teacher's Aids", Computers in Education Summer Conference in Proceedings, Center for Advanced Technology in Education, 1985. pp. 277-283
- Bitter, Gary G., Camuse, Ruth A. Using A Microcomputer in the Classroom, Reston, Virginia: Reston Publishing Company, Inc. 1984.
- Bork, Alfred. Learning With Computers, Digital Press, 1981.
- Brown, Jerald R. Instant BASIC, Beaverton, Oregon: Dilithium Press, 1982.
- Budoff, Milton., Thormann, John., Gras, Ann. Microcomputers in Special Education, Brookline Books, 1984.
- Burns, H. L., and Culp, G. H., "Stimulating Invention in English Composition Through Computer Assisted Instruction." Educational Technology, August, 1980.

- Burke, Robert L. CAI Sourcebook. New Jersey: Prentice-Hall, 1982.
- Burson, Jeanne L. The Author's Guide To Computer Assisted Instruction (CAI). Columbus, Ohio: Division of Computing Services for Medical Education and Research, College of Medicine, The Ohio State University, 1976.
- Callison, Williom L. Using Computer in the Classroom, New Jersey: Prentice-Hall, 1985.
- Chambers, J. A. & Sprecher, J. W. "Computer Assisted Instruction: Current Trends and Critical Issues". Communications of the ACM, 1980. 23(6), pp. 332-342, June 1980.
- Coan, James S. Basic BASIC, Rochelle Park, New Jersey: Hayden Book Company, Inc, 1978.
- Craighead, Donna and Bitter, Gary G., Microcomputers in Education Conference, Computer Science Press, 1984.
- Edward, Judith B. "Computer Software in Educaton". Computers in the Classroom, Washington, D. C. National Education Association, pp. 137-145, 1982.
- Ellis, Allan B. The Use & Misuse of Computers in Education, McGraw-Hill Book Company, 1974.
- Fleit, Lind H. "Computerizing America's Campuses: How Technology is Changing Higher Education", Electronic Learning, March 1987. pp.26
- Gooler, Dennis D. The Education Utility: The Power to Revitalize Education and Society, Educational Technology Publications, 1986.
- Graham, Lyle J. Your IBM PC: A Guide to the IBM Personal Computer, Berkeley, California: Osborne/McGraw-Hill, 1983.
- Grady, M. Tim, and Gawronski, Jane D., Computers in Curriculum and Instruction, Association for Supervision and Curriculum Development, 1983.
- Hallworth, H. J & Brehner, A. Computer Assisted Instruction in Schools: Achievements, Present Developments and Projections for the Future. Edmonton: Alberts Education, 1980.
- Hanfling, Seymour S. "Artificial Intelligence Tutorials", Computers in Education Summer Conference in Proceedings, Center for Advanced Technology in Education, 1985. pp. 59-65

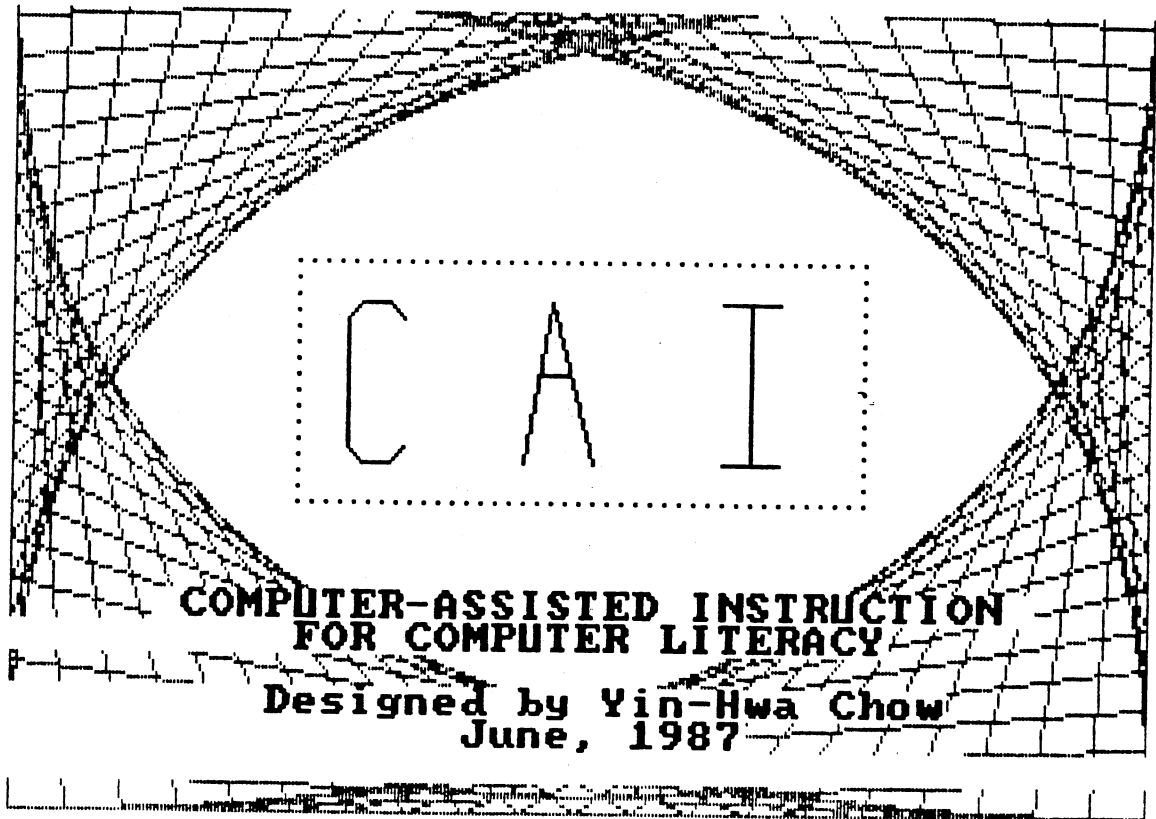
- Heines, Jesse M. Screen Design Strategies for Computer-Assisted Instruction, Digital Press, 1984.
- Hirschbuhl, John J. "Futures: Where will Computer-Assisted Instruction (CAI) Be in 1990?", Educational Technology, Vol.18, No.4, April 1978. pp.62
- Hofmeister, Alan, Microcomputer Applications in the Classroom, Holt, Rinehart and Winston, 1984.
- Johnson, J., and Jongejan, T. "Mathematics and Computers", in Elementary and Secondary School Topics, Association for Computing Machinery, 1981.
- Kepner, Henry S., Jr., Computers in the Classroom, Washington, D. C.: National Education Association, 1982.
- Kinzer, Charlesk., Sherwood, Robert D., Bransford, John D. Computer Strategies for Education: Foundation and Content-Area Application. Merrill Publishing Company, 1986.
- Levien, R. E. The Emerging Technology: Instructional Uses of the Computer in Higher Education. McGraw-Hill, New York, 1972.
- Mandell, Steven L. Complete BASIC Programming, St. Paul, MN: West Publishing Computer, 1987.
- Merrill, Paul F., Tolman, Marvin N., Christensen Larry., Hammons, Kathy., Vincent, Bret R., Reynolds, Peter L. Computers in Education, New Jersey: Prentice-Hall, 1986.
- McGinty, Tony and Reed Sally., Sautter Graig. "Software Trends and Trendsetters" Electronic Learning. January 1987. pp.18-26
- Merrill, M. D. "Learner Control in Computer Based Learning". Computers and Education, 1980. 4(2) pp.77-95
- O'neil, Harold F. JR. Computer-Based Instruction: A State-of-the-Art Assessment, Academic Press, 1981.
- Orwig, Gary W. Creating Computer Programs for Learning. Reston, Virginia: Reston Publishing Company, Inc. A Prentice-Hall Company, 1983.
- Orwig, Gary W., Hodges, William S. The Computer Tutor Winthrop Publishers, Inc., 1982.
- O'shea Tim. Self-Improving Teaching Systems: An Application of Artificial Intelligence to Computer Assisted Instruction, Basel: Birkhauser Verlag, 1979.

- Papert, Seymour. Teaching Children Thinking (Logo Memo 2), MIT Artificial Intelligence Laboratory, Logo Group, Cambridge, MA, October, 1971.
- _____. Mindstorms: Children, Computers, and Powerful Ideas. New York: Basic Books, 1980.
- Price, Wilson T. Elements of BASIC-Plus, Programming, Holt-rinehart and Winston, 1982.
- Ragsdale, Ronald G. Computers in the Schools: A Guide for Planning, Toronto, Ontario: The Ontario institute for studies in education, 1982.
- Reinhold, Fran. "Computing in America", Electronic Learning, October 1986. pp.26
- Sadowski, Barbara R. "One Solution to the Courseware Crisis". 1982-1984 Microcomputers in Education Conference. Rockville, Maryland: Computer Science Press, 1984. pp.143-147
- Smith, P. R. Advances in Computer-Assisted Learning: Selected Proceedings from the CAL 85 Symposium, Pergamon Press, 1985.
- Steinberg, Esther R. Teaching Computers to Teach. Hillsdale, New Jersey: Lawrence Erlbaum Associates, Publishers, 1984.
- Sharpe, William F. and Jacob, Nancy. BASIC: An Introduction to Computer Programming Using the BASIC Language, New York: The Free Press, 1971.
- Tashner, John H. Computer Literacy for Teachers: Issues, Questions, and Concerns. Phoenix, Arizona: The Orxy Press, 1984.
- Taylor, Robert P. The Computer in the School: Tutor, Tool, Tutee, Teachers College Press, 1980.
- Williams, Christopher. "Trendsetter AI Rogers: Software Earns Him Respect, Not Money." Electronic Learning. January, 1987. pp.19
- Walker, Decker F., and Hess, Robert D., Instructional Software, Wadsworth Publishing Company, Belmont, California, 1984,

APPENDIXES

APPENDIX A

THE RESULT OF DRILL INSTRUCTION
FOR SELECTED LESSON



**COMPUTER-ASSISTED INSTRUCTION
FOR COMPUTER LITERACY**

**Designed by Yin-Hwa Chow
June, 1987**

```
*****  
*                                     *  
*   WELCOME TO YIN-HWA SOFTWARE COMPANY   *  
*                                     *  
*****
```

This software package is designed for a course on computer literacy. It has been divided into two parts,

- (A) Computers today
- (B) BASIC programming.

Part (A) consists of six lessons with 120 drill problems

Part (B) also consists of six lessons. This part is designed to be a tutorial lesson for BASIC programming. Each lesson provides five problems for a learning check.

ENTER YOUR NAME TO BEGIN

? Linda

Here are some special pointers that you need to remember:

1. Any time you respond to a computer question, you should press the ENTER key (<---) and the computer will accept the data for you.
2. Be sure to answer the computer questions exactly as prompted. Follow the directions carefully. (For example: key Y for yes and N for no. Do not key in yes when Y is requested.)
3. Be sure you consult with your teacher when you don't understand some terminology or how the equipment works
4. If you have access to a printer, print out your screen using Shift and PrtSc keys pressed simultaneously.
5. If you want to EXIT while running the program, press the Ctrl and Break keys simultaneously.

PRESS ANY KEY FOR PROCESSING

MAIN MENU

COMPUTER LITERACY
~~~~~

PART &lt; A &gt;. COMPUTERS TODAY

PART &lt; B &gt;. BASIC PROGRAMMING

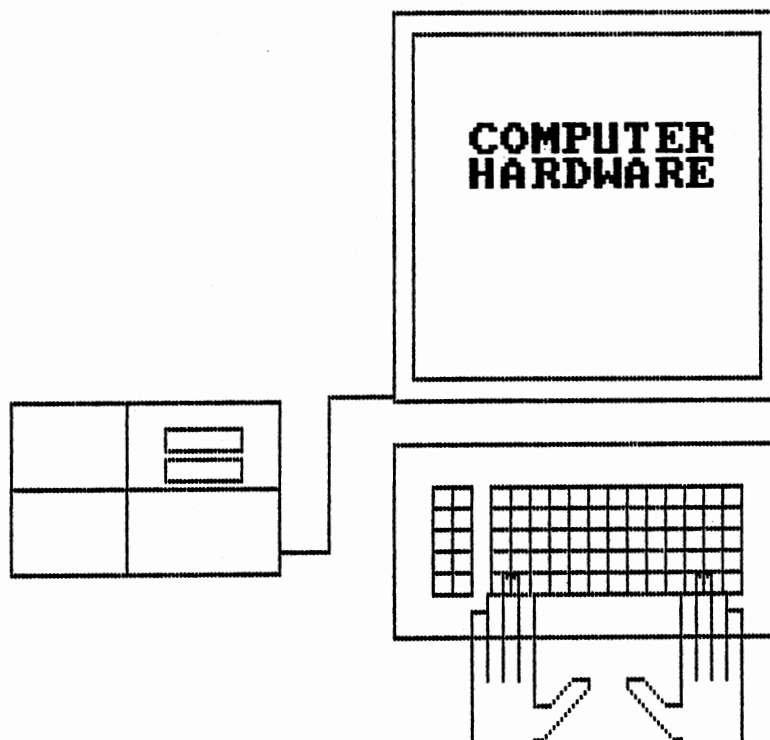
SELECT A or B ? A

**PART A****COMPUTERS TODAY****MENU**  
-----

- <1>. COMPUTER TERMS
- <2>. COMPUTER CONCEPTS
- <3>. COMPUTER HARDWARE SYSTEMS
- <4>. COMPUTER SOFTWARE
- <5>. INFORMATION SYSTEMS
- <6>. COMPUTERS AND SOCIETY

YOUR CHOICE (1 - 6) ? 3

processing, please WAIT.....

**EXERCISES III**

The multiple-choice questions which follow will help test and reinforce your understanding of the subject: COMPUTER HARDWARE.

You will be given 20 questions each worth 5 points. After each question, you will see a "? " prompt. Type your answer, then press the RETURN key.

Each answer, whether right or wrong, contains some additional facts to enhance your knowledge.

You are allowed two attempts for each question. If your first answer is incorrect, please wait and do not press any key. The screen will be cleared and the same question will reappear for the second time.

Upon finishing the 20 questions, you will be given the number you answered correctly and your score.

GOOD LUCK ! Linda

PRESS ANY KEY TO CONTINUE

1. Which of the following is not a characteristic of data entry devices used in direct-access processing:
- a) They create a direct interaction between people and computers
  - b) They are economical even when input data volume is low or irregular
  - c) They require the use of data recording media
  - d) They are generally located at the data source

? c

**THAT'S GREAT! Linda, YOU ARE OFF TO AN EXCELLENT START!**

**PRESS ANY KEY TO CONTINUE**



2. Teleprinter terminals

- a) are all intelligent devices
- b) are of very recent origin
- c) cannot be portable
- d) have keyboards, printers, and the ability to communicate with distant CPUs

? d

THAT'S GREAT! Linda , KEEP ON GOING !

PRESS ANY KEY TO CONTINUE

3. POS terminals

- a) are general-purpose devices capable of handling more applications than teleprinter terminals
- b) are used only in supermarkets
- c) may use optical scanners to read tags, bar codes, and credit cards
- d) cannot be used in conjunction with financial transaction devices

? d

SORRY, INCORRECT !!!

It can be used to update the customer's credit account, inventory records and provide sales analysis information to managers.

PLEASE TRY AGAIN.

## 3. POS terminals

- a) are general-purpose devices capable of handling more applications than teleprinter terminals
- b) are used only in supermarkets
- c) may use optical scanners to read tags, bar codes, and credit cards
- d) cannot be used in conjunction with financial transaction devices

? a

SORRY, INCORRECT !!!

Linda , Point-of-sale terminals are special purpose devices used in supermarkets and retail stores.

The answer is:

(c)

PRESS ANY KEY TO CONTINUE

4. An automated teller machine

- a) is operated by a skilled teller
- b) transmits data coded on a plastic card to the CPU
- c) can be used only during banking hours
- d) reads data from a microcomputer chip embedded on a plastic card

? c

SORRY, INCORRECT !!!

Linda , It can be used 24 hours a day.

PLEASE TRY AGAIN.

4. An automated teller machine

- a) is operated by a skilled teller
- b) transmits data coded on a plastic card to the CPU
- c) can be used only during banking hours
- d) reads data from a microcomputer chip embedded on a plastic card

? d

SORRY, INCORRECT !!!

Linda , It reads data from a card with strip of tape to a CPU which activates a customer's account.

The answer is:

(b)

PRESS ANY KEY TO CONTINUE

5. A visual display terminal

- a) is, by definition, a dumb terminal
- b) can process either graphic or alphanumeric capabilities,  
but not both
- c) must be located at the site of the CPU
- d) is the most popular input device used today in direct-access processing

? c

SORRY, INCORRECT !!!

Linda , It transmits the data through a workstation.

PLEASE TRY AGAIN.

5. A visual display terminal

- a) is, by definition, a dumb terminal
- b) can process either graphic or alphanumeric capabilities,  
but not both
- c) must be located at the site of the CPU
- d) is the most popular input device used today in direct-access processing

? d

THAT'S GREAT! Linda, THIS WAS A TOUGH ONE.

PRESS ANY KEY TO CONTINUE

6. A visual-display terminal generally does not receive input data from

- a) a felt-tip pen
- b) an input tablet
- c) a keyboard
- d) a light pen

? c

SORRY, INCORRECT !!!

Linda , Keyboard is the basic input device.

PLEASE TRY AGAIN.



6. A visual-display terminal generally does not receive input data from

- a) a felt-tip pen
- b) an input tablet
- c) a keyboard
- d) a light pen

? a

GOOD JOB ! Linda, LET'S DO THE NEXT ONE

PRESS ANY KEY TO CONTINUE

7. Computer graphic techniques

- a) have declined in popularity in recent years
- b) are currently being used in the design of airplanes and buildings but they are of little use to business decision makers
- c) have increased the time required to complete final engineering drawings
- d) make it possible for a designer to make quick changes and modifications to preliminary drawings

? b

**SORRY, INCORRECT !!!**

Linda , They are also being used to present business financial statements and operating data in graphic form.

**PLEASE TRY AGAIN.**

7. Computer graphic techniques

- a) have declined in popularity in recent years
- b) are currently being used in the design of airplanes and buildings but they are of little use to business decision makers
- c) have increased the time required to complete final engineering drawings
- d) make it possible for a designer to make quick changes and modifications to preliminary drawings

? d

NICE JOB !!

PRESS ANY KEY TO CONTINUE

**8. An intelligent terminal**

- a) has a microprocessor, but it cannot be programmed by the user
- b) can be used for small data processing jobs without the need to interact with a large CPU
- c) carries on a conversation in ENGLISH with the user
- d) cannot be used to edit data originating at dumb terminal

? b

**EXCELLENT JOB !Linda , YOU ARE REALLY ON A ROLL!**

**PRESS ANY KEY TO CONTINUE**

9. Voice input systems

- a) must be speaker-dependent
- b) must be speaker-independent
- c) have limited vocabularies and can't accept words spoken in a continuous stream
- d) are able to take dictation and produce finished letters

? b

SORRY, INCORRECT !!!

Linda , It also can be speaker-dependent.

PLEASE TRY AGAIN.

9. Voice input systems

- a) must be speaker-dependent
- b) must be speaker-independent
- c) have limited vocabularies and can't accept words spoken in a continuous stream
- d) are able to take dictation and produce finished letters

? c

GOOD JOB ! Linda, LET'S DO THE NEXT ONE

PRESS ANY KEY TO CONTINUE

10. Which of the following devices is not for direct-access processing?

- a) portable data entry terminals
- b) floppy disks
- c) visual display terminals
- d) automated teller machine

? b

THAT'S GREAT! Linda , KEEP ON GOING !

PRESS ANY KEY TO CONTINUE

11. Which of the following is not a device in a computer system?

- a) output unit
- b) input unit
- c) arithmetic-logic unit
- d) CPU

? c

EXCELLENT JOB !Linda , YOU ARE REALLY ON A ROLL!

PRESS ANY KEY TO CONTINUE



12. Which of the following cannot serve as an output device?

- a) card punch
- b) printer
- c) magnetic type drive
- d) CPU

? a

SORRY, INCORRECT !!!

Linda , Card punch is an output device.

PLEASE TRY AGAIN.

12. Which of the following cannot serve as an output device?

- a) card punch
- b) printer
- c) magnetic type drive
- d) CPU

? b

SORRY, INCORRECT !!!

Linda , Printer is an output device.

The answer is:

(d)

PRESS ANY KEY TO CONTINUE

13. The central processing unit controls:

- a) reading of input from an input device
- b) processing of input data
- c) all arithmetic operations
- d) all of the above

? d

PRETTY GOOD !      I KNEW YOU COULD DO IT!

PRESS ANY KEY TO CONTINUE

14. Which of the following is not a characteristic of batch processing?

- a) computer operations are performed periodically
- b) not especially timely processing
- c) inquiries can be answered on a timely and effective basis
- d) off-line operations may be used in conjunction with batch processing

? a

SORRY, INCORRECT !!!

Linda , It is a characteristic of batch processing.

PLEASE TRY AGAIN.

14. Which of the following is not a characteristic of batch processing?
- a) computer operations are performed periodically
  - b) not especially timely processing
  - c) inquiries can be answered on a timely and effective basis
  - d) off-line operations may be used in conjunction with batch processing

? b

SORRY, INCORRECT !!!

Linda , It is a characteristic of batch processing.

The answer is:

(c)

PRESS ANY KEY TO CONTINUE

15. Characteristics of COM include all but one of the following:

- a) output is typically produced in an interactive mode
- b) output is produced at very great speeds
- c) output is produced in compact form
- d) for large-volume jobs, microfilm is cheaper than paper

? a

YOU GOT IT !      LET'S CONTINUE

PRESS ANY KEY TO CONTINUE

16. Which of the following is not true about serial printers?

- a) relatively slow
- b) used as an interactive terminal
- c) relatively inexpensive
- d) generally uses electrostatic technology

? a

SORRY, INCORRECT !!!

Linda , This is true about serial printers.

PLEASE TRY AGAIN.

16. Which of the following is not true about serial printers?

- a) relatively slow
- b) used as an interactive terminal
- c) relatively inexpensive
- d) generally uses electrostatic technology

? b

SORRY, INCORRECT !!!

Linda , This is true about serial printers.

The answer is:

(d)

PRESS ANY KEY TO CONTINUE



17. Soft copy output is generally produced by a:

- a) card punch
- b) key punch
- c) CRT
- d) Typewriter terminal

? c

THAT'S GREAT! Linda , KEEP ON GOING !

PRESS ANY KEY TO CONTINUE

18. Which of the following is not a feature of tape and disk?

- a) frequently used for storing master files
- b) can store and access records sequentially
- c) function effectively in an on-line, real-time mode
- d) records are frequently blocked to maximize efficient use

? a

SORRY, INCORRECT !!!

Linda , This statement is true.

PLEASE TRY AGAIN.

18. Which of the following is not a feature of tape and disk?

- a) frequently used for storing master files
- b) can store and access records sequentially
- c) function effectively in an on-line, real-time mode
- d) records are frequently blocked to maximize efficient use

? b

SORRY, INCORRECT !!!

The answer is:

(c)

PRESS ANY KEY TO CONTINUE

19. Which of the following is not a characteristic of disk packs?

- a) consists of numerous disks
- b) each individual disk has numerous surfaces
- c) each disk has numerous tracks
- d) each disk has numerous sectors

? b

THAT'S GREAT! Linda, THIS WAS A TOUGH ONE.

PRESS ANY KEY TO CONTINUE

20. Which of the following is not a characteristic of tape?

- a) densities can be measured in hundreds or thousands of bits per inch
- b) physical records are grouped into logical ones that are separated by IBGs
- c) records can be fixed length or variable length
- d) data can be written on tape using a key device

? a

SORRY, INCORRECT !!!

Linda , This is a characteristic of tape.

PLEASE TRY AGAIN.

20. Which of the following is not a characteristic of tape?

- a) densities can be measured in hundreds or thousands of bits per inch
- b) physical records are grouped into logical ones that are separated by IBGs
- c) records can be fixed length or variable length
- d) data can be written on tape using a key device

? b

YOU GOT IT !

PRESS ANY KEY TO CONTINUE

**You are finished.**

**Linda, YOU HAVE ANSWERED 12 QUESTIONS CORRECTLY OUT OF 20 QUESTIONS!!**

**YOU HAD 60 % CORRECT ANSWERS!**

**If you are unsure of some of the questions, please ask your instructor to explain them.**

**If your score was less than 80%, you should review this lesson.**

**PRESS ANY KEY TO CONTINUE**

-----  
**WHAT'S NEXT**  
-----

You have just completed this lesson. You now have three options:

1. Review this lesson
2. Try a new lesson
3. Exit

( Select 1-3 )

? 2

processing please WAIT.....



APPENDIX B

THE RESULT OF TUTORIAL INSTRUCTION  
FOR SELECTED LESSON

## MAIN MENU

COMPUTER LITERACY  
~~~~~

PART < A >. COMPUTERS TODAY

PART < B >. BASIC PROGRAMMING

SELECT A or B ? B

PART B

BASIC PROGRAMMING

MENU

- <1>. GETTING STARTED
- <2>. INPUT AND OUTPUT
- <3>. CONTROL STATEMENT
- <4>. LOOPING AROUND
- <5>. SUBROUTINES AND FUNCTIONS
- <6>. ARRAYS

YOUR CHOICE (1 - 6) ? 3

processing, please WAIT.....

CCCCC	OOOOO	NNNNN	TTTTT	RRRRR	OOOOO	L
C	O O	N N	T	R R	O O	L
C	O O	N N	T	RRRRR	O O	L
C	O O	N N	T	R R	O O	L
CCCCC	OOOOO	N N	T	R R	OOOOO	LLLLL

SSSSS	TTTTT	A	TTTTT	EEEE	MMMM	EEEE	NNNN	TTTTT	SSSS
S	T	A A	T	E	M M M	E	N N	T	S
SSSSS	T	AAAAA	T	EEEE	M M M	EEEE	N N	T	SSSSS
S	T	A A	T	E	M M M	E	N N	T	S
SSSSS	T	A A	T	EEEE	M M M	EEEE	N N	T	SSSSS

This lesson introduces the CONTROL STATEMENTS, which allows the programmer to control the order of the program.

GOTO, IF/THEN/ELSE, and ON/GOTO are control statements introduced here

Control statements are also called conditional branching. Branching is what happens when the computer follows an instruction that sends it to a different part of the program, not in regular line number order.

PRESS ANY KEY TO CONTINUE

GO TO (GOTO)

Look at the output of this program.

10 GOTO 30	-----> The GOTO statement tells the computer
20 PRINT "A+B"	go to line 30, which is the END (stop).
30 END	
RUN	-----> So, there is no output.

PRESS ANY KEY TO CONTINUE

Here is another program with a GOTO statement:

```
10 A=5
20 B=4
30 GOTO 50
40 PRINT A
50 PRINT B
60 END
RUN
4
```

-----] This statement tells the computer
to skip line 40 to line 50.
<-----]

So, the output only executes line 50
(to display the value of the B, which is 4).

GOTO statement tells the computer to go out of its regular order.

PRESS ANY KEY TO CONTINUE

What is the capital of Arizona?
a. Tucson
b. Tempe
c. Phoenix
d. Austin

? a
SORRY, INCORRECT!

How did the computer decide whether the user's answer was right?
The following program and output will show you how it works.

```
10 PRINT "What is the capital of Arizona?"
20 PRINT "a. Tucson"
30 PRINT "b. Tempe"
40 PRINT "c. Phoenix"
50 PRINT "d. Austin"
20 INPUT MS
30 IF MS="c" THEN PRINT "YOU GOT IT"
40 IF MS<>"c" THEN PRINT "SORRY, INCORRECT!"
50 END
```

```
RUN
What is the capital of Arizona?
a
SORRY, INCORRECT!
```

PRESS ANY KEY TO CONTINUE

IF/THEN

The general format of IF/THEN statement is:

Line# IF condition THEN line#
(statement)

For example:

100 IF A=2 THEN 300

The following are relational symbols that can be used:

SYMBOL	CONDITION	EXAMPLE
<	less than	A < B
>	Greater than	A > B
=	Equal to	A = B
<=	Less than or equal to	A <= B
>=	Greater than or equal to	A >= B
<>	Not equal to	A <> B

PRESS ANY KEY TO CONTINUE

Before we show you more detail of the IF/THEN statement, we would like to introduce the FLOWCHART SYMBOLS.

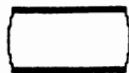
A FLOWCHART is a drawing of the steps used to solve a problem. It shows the actual flow of the logic of the program.

A FLOWCHART is helpful in understanding how the computer follows the instructions of the program.

It shows the path of computer takes when following instructions.

PRESS ANY KEY TO CONTINUE

FLOWCHART SYMBOLS



TERMINAL SYMBOL indicates start or end of the program.



INPUT/OUTPUT SYMBOL represents either inputting data (includes READ/DATA) or outputting results.



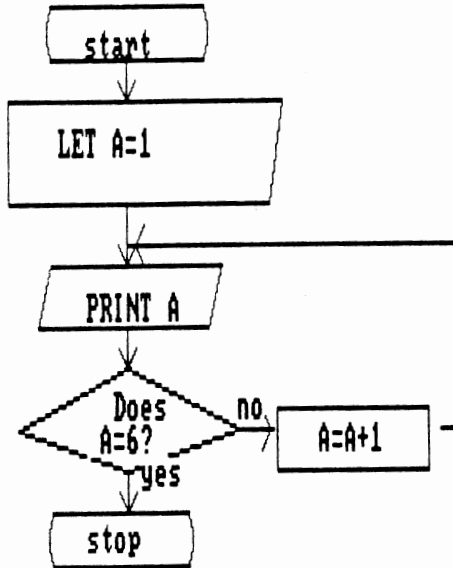
PROCESSING SYMBOL represents calculations or other processing operations.



DECISION SYMBOL is used whenever a decision is to be made.

PRESS ANY KEY TO CONTINUE

HERE IS A PROGRAM WITH IF/THEN STATEMENT
AND A FLOWCHART:



```
10 LET A=1
20 PRINT A;
30 IF A=6 THEN 100
40 A=A+1
50 GOTO 20
100 END
```

RUN

1 2 3 4 5 6

PRESS ANY KEY TO CONTINUE

Look at this program and the output:

```
10 PRINT "HOW OLD ARE YOU?"
20 INPUT M
30 IF M=16 THEN PRINT "YOU ARE OLD ENOUGH TO DRIVE."
40 IF M>16 THEN PRINT "YOU ARE OLD ENOUGH TO DRIVE."
50 IF M<16 THEN PRINT "YOU ARE NOT OLD ENOUGH TO DRIVE."
60 END
```

THE OUTPUT WILL DEPEND ON THE USER'S ANSWER:

User 1.

```
HOW OLD ARE YOU?
?16
YOU ARE OLD ENOUGH TO DRIVE.
```

User 2.

```
HOW OLD ARE YOU?
?23
YOU ARE OLD ENOUGH TO DRIVE.
```

User 3.

```
HOW OLD ARE YOU?
?14
YOU ARE NOT OLD ENOUGH TO DRIVE.
```

PRESS ANY KEY TO CONTINUE

Let's compare these two programs:

A.

```
10 PRINT "What is the capital of Texas?"
60 INPUT M$
70 IF M$="AUSTIN" THEN PRINT "YOU GOT IT"
80 IF M$<>"AUSTIN" THEN PRINT "SORRY, INCORRECT!"
90 END
```

```
RUN
What is the capital of Texas?
AUSTIN
YOU GOT IT!
```

B.

```
10 PRINT "What is the capital of Texas?"
20 INPUT M$
30 IF M$="AUSTIN" THEN 40 ELSE 50
40 PRINT "YOU GOT IT!" : GOTO 60
50 PRINT "SORRY, INCORRECT!"
60 END
```

```
RUN
What is the capital of Texas?
AUSTIN
YOU GOT IT!
```

The two programs outputs are the same.
But, line 30 of program B has a new statement IF/THEN/ELSE.

PRESS ANY KEY TO CONTINUE

IF/THEN/ELSE

The general format of the IF/THEN/ELSE statement is:

```
Line# IF condition THEN clause ELSE clause
                (line#)      (line#)
```

For example:

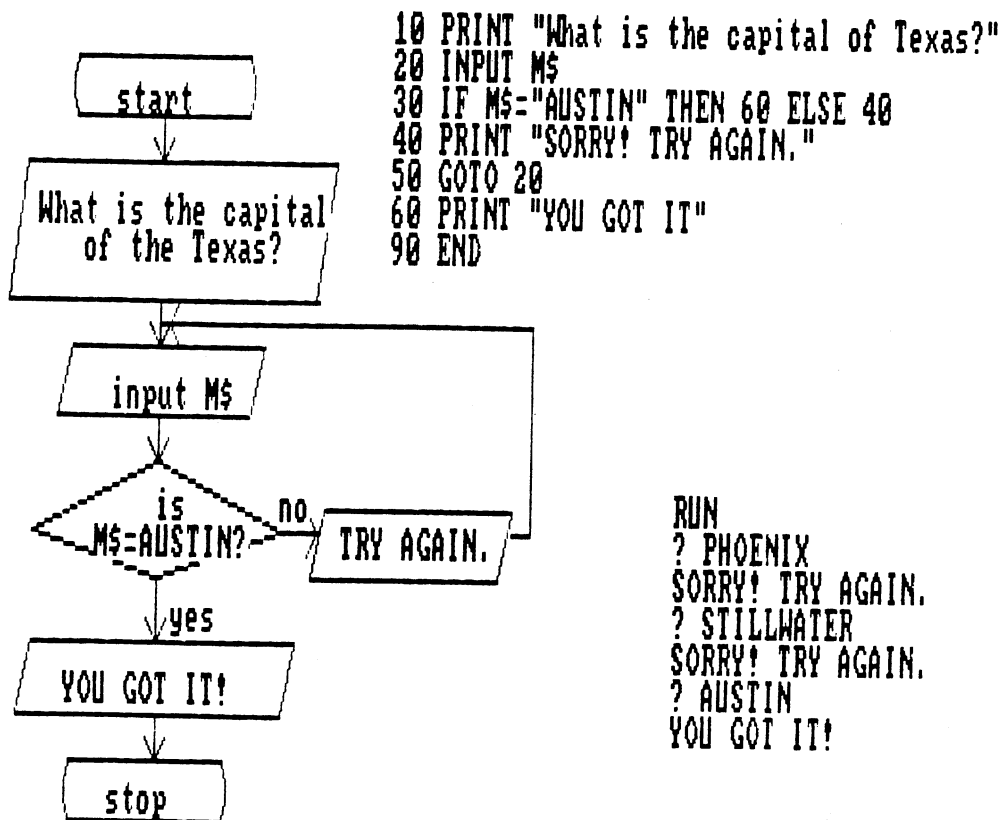
```
10 IF GPA > 3.5 THEN 50 ELSE 60
50 PRINT "GOOD"
60 PRINT "NEEDS IMPROVEMENT!"
```

OR

```
10 IF GPA > 3.5 THEN PRINT "GOOD "ELSE PRINT "NEEDS IMPROVEMENT!"
```

A sample program with IF/THEN/ELSE statement and its flowchart are given in the next screen.

PRESS ANY KEY TO CONTINUE



PRESS ANY KEY TO CONTINUE

ON/GOTO

The ON/GOTO statement general format is:

Line# ON expression GOTO line#1,line#2,line#3,...,line#n

Statement	Computer Execution
10 ON A GOTO 30,50,100	If A=1, goto line 30 If A=2, goto line 50 If A=3, goto line 100
60 ON B/50 GOTO 150,200	If B/50=1, goto line 150 If B/50=2, goto line 200

PRESS ANY KEY TO CONTINUE

PRACTICE

1. Which of the following is not a valid IF/THEN statement?
- a. 40 IF B=2 THEN 150
 - b. 20 IF A <> "NO "THEN 60
 - c. 100 IF Y\$="YES" THEN 150
 - d. 130 IF F\$="FALSE" THEN 200

? b

THAT'S GREAT! Linda, YOU ARE OFF TO AN EXCELLENT START!

PRESS ANY KEY TO CONTINUE

2. What statement is supposed be in line 30?

```
10 B = 2
20 PRINT B;
30 IF
40 LET B = B + 2
50 GOTO 20
60 END
RUN
2 4 6 8 10 12 14 16 18 20
```

- a. IF B=20 THEN 60
- b. IF B>20 THEN 60
- c. IF B<20 THEN 50
- d. IF B=20 THEN 50

? c

SORRY, INCORRECT! Linda, THE CORRECT ANSWER IS 'a'

PRESS ANY KEY TO CONTINUE

3. What is missing in line 40?

```
10 A=100
20 PRINT A;
30 IF A=0 THEN 60
40
50 GOTO 20
60 END
RUN
100 90 80 70 60 50 40 30 20 10 0
```

- a. A-10=A
- b. PRINT A-10
- c. A=A-10
- d. PRINT "A-10"

? c

EXCELLENT JOB !Linda , YOU ARE REALLY ON A ROLL!

PRESS ANY KEY TO CONTINUE

4. What is the output for the following program?

```
10 LET G = 1
20 IF G>6 THEN 100
30 PRINT G ;
40 LET X=X+1
50 GOTO 20
60 END
```

- a. 123456
- b. 1 2 3 4 5 6 7
- c. 1 2 3 4 5 6
- d. 1234567

? d

SORRY! The correct answer is 'c'

PRESS ANY KEY TO CONTINUE

5. In an IF/THEN/ELSE statement, the clause following the THEN or ELSE may be:
- a. a line number
 - b. a single BASIC statement
 - c. multiple BASIC statement
 - d. all of the above

? d

PRETTY GOOD ! I KNEW YOU COULD DO IT!

PRESS ANY KEY TO CONTINUE

WHAT'S NEXT ?

You have just completed this lesson. You now have three options:

1. Review this lesson
2. Try a new lesson
3. Exit

(Select 1-3)

? 2

WAIT FOR PROCESSING

MAIN MENU**COMPUTER LITERACY**
~~~~~**PART < A >. COMPUTERS TODAY****PART < B >. BASIC PROGRAMMING****SELECT A or B ? B**



## PART B

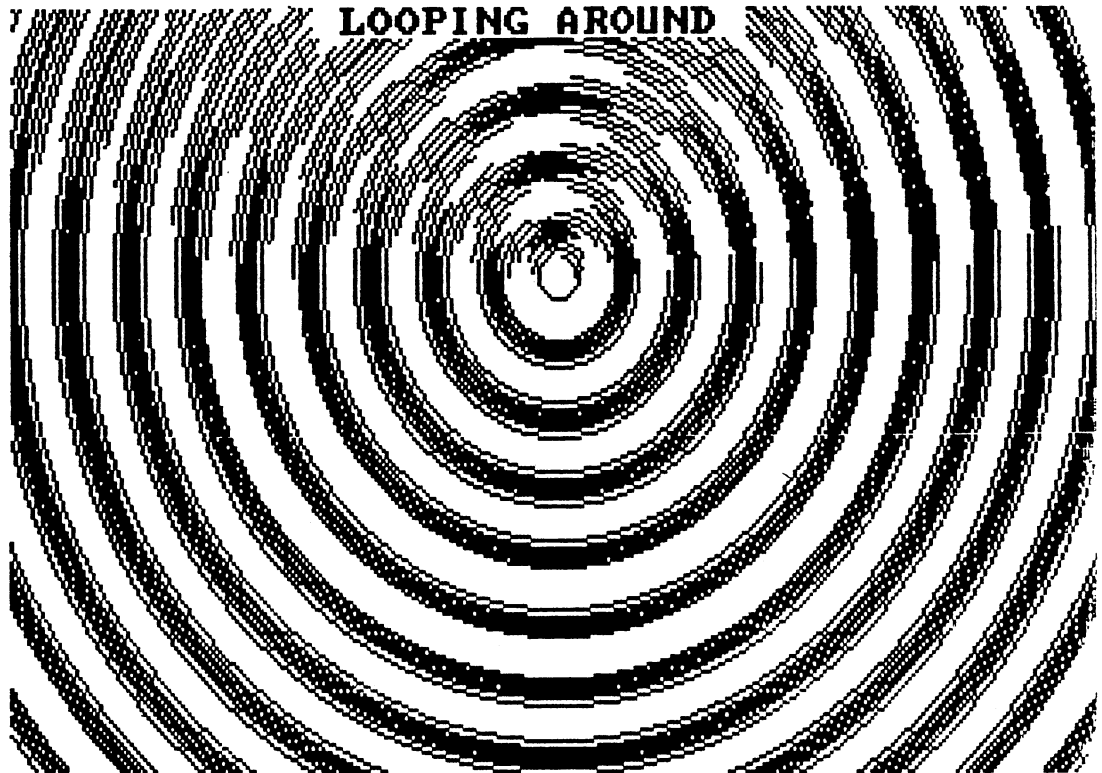
## BASIC PROGRAMMING

MENU  
-----

- <1>. GETTING STARTED
- <2>. INPUT AND OUTPUT
- <3>. CONTROL STATEMENT
- <4>. LOOPING AROUND
- <5>. SUBROUTINES AND FUNCTIONS
- <6>. ARRAYS

YOUR CHOICE (1 - 6) ? 4

processing, please WAIT.....



You have used the control statement to make a loop in a program. A computer loop may be defined as a self-repeating sequence of program statements.

In this lesson, you will learn how to tell the computer exactly how many loops you want the computer to make. Also, you will learn how to use FOR/NEXT statements to make loops in an efficient manner.

PRESS ANY KEY TO CONTINUE

FOR/NEXT  
-----

FOR and NEXT always appear together. If you have a FOR, you must have a NEXT in the program.

Look at these two program:

A.

```
10 PRINT "NEVER STOP"  
20 GOTO 10  
30 END  
RUN  
NEVER STOP  
NEVER STOP  
NEVER STOP  
NEVER STOP  
NEVER STOP  
NEVER STOP
```

B.

```
10 FOR L=1 TO 4  
20 PRINT "LOOP"; L  
30 NEXT L  
40 END  
RUN  
LOOP 1  
LOOP 2  
LOOP 3  
LOOP 4
```

The output of program A will go on forever. But, program B has a FOR/NEXT statement which tells the computer to make 4 loops.

PRESS ANY KEY TO CONTINUE

Let's trace this program.

```
10 FOR L=1 TO 4
20 PRINT "LOOP"; L
30 NEXT L
40 END
RUN
  LOOP 1
  LOOP 2
  LOOP 3
  LOOP 4
```

LOOP

The first time through the loop, L is 1.  
So, the computer prints LOOP 1.  
The second time through the loop, L is 2.  
So, the computer prints LOOP 2, and so on.

Line 10 begins the loop by storing a 1 in L.

Line 20 tells the computer to print LOOP and the number which stored in L.

Line 30 sends the computer back to the line 10.

Now line 10 puts a 2 in L.

Look at line 20 carefully. Each time the computer goes through the loop, it prints the number stored in L.

Then, at the same time it prints the word LOOP.

PRESS ANY KEY TO CONTINUE

Compare the following two programs:

A.

```
10 LET A=1
20 PRINT A;
30 IF A=6 THEN 100
40 A=A+1
50 GOTO 20
60 END
RUN
1 2 3 4 5 6
```

B.

```
10 FOR A = 1 TO 6
20 PRINT A;
30 NEXT A
40 END
RUN
1 2 3 4 5 6
```

Program B only uses 4 statements to produce the same output as program A, because Program B uses a FOR/NEXT statement instead of a control statement.

PRESS ANY KEY TO CONTINUE

Here is another program that uses a FOR/NEXT statement:

```
10 FOR A = 10 TO 1 STEP -2
20 PRINT A;
30 NEXT A
40 END
RUN
  10 8 6 4 2
```

STEP means the value to be increased.

Since the step is a NEGATIVE number, the initial value (10) should be greater than the terminal value (1).

First loop A = 10

Second loop A = 8

Third loop A = 6

Fourth loop A = 4

Fifth loop A = 2

The terminal value is 1, so the computer stops executing.

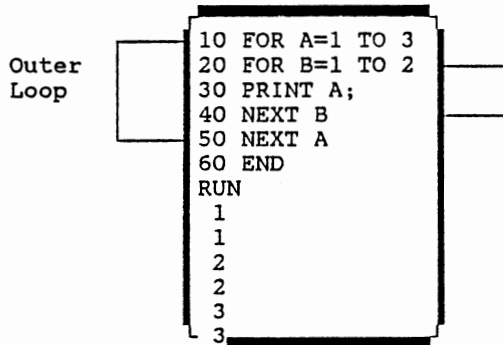
PRESS ANY KEY TO CONTINUE

### NESTED FOR/NEXT

---

Nested loops can be done with FOR/NEXT statements.

For example:



Line 10 stores the value 1 to A  
 Line 20 stores the value 1 to B  
 Line 30 tells the computer  
 print the value of A which is 1  
 Line 40 tells the computer to  
 go back to line 20.  
 The value of A does not change.  
 So, the computer still prints 1.  
 Now, line 50 tells the computer to  
 go back to the line 10.  
 The loop run starts over again.

The computer first processes  
 the INNER LOOP.  
 Then, processes the OUTER LOOP.

PRESS ANY KEY TO CONTINUE



The following statements will show you the rules of using nested FOR/NEXT loops

1. Each loop must have unique loop control variable:

INCORRECT

```
A = 1 TO 3
A = 2 TO 8
NEXT A
NEXT A
```

CORRECT

```
A = 1 TO 3
B = 2 TO 8
NEXT B
NEXT A
```

2. Each loop can not overlap each other:

INCORRECT

```
A = 1 TO 3
B = 2 TO 8
NEXT A
NEXT B
```

CORRECT

```
A = 1 TO 3
B = 2 TO 8
NEXT B
NEXT A
```

PRESS ANY KEY TO CONTINUE

### MULTIPLE NESTED FOR/NEXT

---

It is possible to nest many loops within one another.

For example:

```
FOR A = 1 TO 5
  PRINT "FUN" ;
  FOR B = 1 TO 1
    PRINT "IS" ;
    FOR C = 1 TO 1
      PRINT "COMPUTER " ;
      NEXT C
    NEXT B
  NEXT A
```

PRESS ANY KEY TO CONTINUE

Compare the following two programs:

A.

```

Loop 1  ┌── FOR A = 1 TO 5
        │   PRINT "COMPUTER ";
        │
        │   Loop 2 ┌── FOR B = 1 TO 1
        │           │   PRINT "IS ";
        │           │
        │           │   Loop 3 ┌── FOR C = 1 TO 1
        │           │           │   PRINT "FUN"
        │           │           │   NEXT C
        │           │           └── NEXT B
        │           └── NEXT A
        └── NEXT A
  
```

```

RUN
COMPUTER IS FUN
COMPUTER IS FUN
COMPUTER IS FUN
COMPUTER IS FUN
COMPUTER IS FUN
  
```

B.

```

Outer Loop ┌── FOR A=1 TO 2
            │   FOR B = 1 TO 2 ─── Inner Loop
            │   PRINT A "*"B "= "A * B
            │   NEXT B
            └── NEXT A
  
```

```

RUN
1 x 1 = 1
1 x 2 = 2
2 x 1 = 2
2 x 2 = 4
  
```

Program A shows that each loop is completely within its outer loop.

PRESS ANY KEY TO CONTINUE

## PRACTICE

-----

1. When the terminal value is exceeded in a FOR/NEXT loop, control is transferred to the
  - a. statement immediately following the NEXT statement.
  - b. END statement.
  - c. statement preceding the FOR statement.
  - d. next FOR statement.

? b

SORRY, Linda , THE ANSWER IS 'a'

PRESS ANY KEY TO CONTINUE

2. How many times would a loop containing the following statement be executed?

```
100 FOR I = 40 TO 22 STEP -5
```

- a. two times
- b. three times
- c. four times
- d. five times

? d

SORRY! The correct answer is 'c'

PRESS ANY KEY TO CONTINUE

3. What is the output from the following program?

```
10 X=20
20 FOR X = 1 to 5
30 PRINT X
40 NEXT X
50 END
RUN
```

a. 1 2 3 4 5

b. 1  
2  
3  
4  
5

c. 20 21 22 23 24

d. 20  
21  
22  
23  
24

? b

GOOD JOB ! Linda, LET'S DO THE NEXT ONE

PRESS ANY KEY TO CONTINUE

4. How many times will the following loops be executed?  
10 FOR P = 1 TO 12 STEP 2  
20 FOR Q = 1 TO 2 STEP .5  
30 PRINT P,Q  
40 NEXT Q  
50 NEXT P  
60 END

- a. 3
- b. 6
- c. 18
- d. 12

? c

EXCELLENT JOB !Linda , YOU ARE REALLY ON A ROLL!

PRESS ANY KEY TO CONTINUE

5. Which of the following is a correct nested FOR/NEXT loop?

a. FOR I=1 to 10  
FOR J=2 TO 8  
NEXT I  
NEXT J

b. FOR I=1 to 5 step -1  
FOR J=1 to 4  
NEXT J  
NEXT I

c. FOR I=1 to 6  
FOR I=2 TO 57  
NEXT I  
NEXT I

d. FOR V=1 to 30  
FOR Q=2 to 6  
NEXT Q  
NEXT V

? b

SORRY! The correct answer is 'd'

PRESS ANY KEY TO CONTINUE



-----  
WHAT'S NEXT ?  
-----

You have just completed this lesson. You now have three options:

1. Review this lesson
2. Try a new lesson
3. Exit

( Select 1-3 )

? 3

THANK YOU FOR YOUR ATTENTION.

GOOD BYE! HAVE A NICE DAY ! Linda

A>

VITA

Yin-Hwa Chow

Candidate for the Degree of  
Master of Science

Thesis: A COMPUTER-ASSISTED INSTRUCTION SOFTWARE PACKAGE FOR  
COMPUTER LITERACY: DESIGN AND DEVELOPMENT

Major Field: Business Education

Biographical:

Personal Data: Born in Shin-Chu, Taiwan, Republic of  
China, May 13, 1955, the daughter of Ming-Sing Chow  
and Yin-Pu Chow.

Education : Graduated from Shin-Chu High School, Shin-  
Chu, Taiwan, in June, 1973; Received Bachelor of  
Laws degree in Economics from Chinese Culture  
University, Taipei, Taiwan, in 1978; Completed  
requirements for the Master of Science degree in  
Business Education at Oklahoma State University in  
July, 1988.

Professional Experience: High School Teacher, Kung-Fu  
Business High School, Taiwan, Republic of China,  
1978-1981; Instructor and administrative assistant  
in Computer Center, Tai-Hwa Industrial Junior  
College, Taiwan, Republic of China, 1981-1984;  
Graduate Research Assistant, College of Business,  
Oklahoma State University, 1986-1987.

Professional Organizations: Society of Delta Pi Epsilon.