Order Number 9504240

The effects of student aptitude and presentation strategies on computer anxiety and student achievement of computer skills

Leso, Thomas Joseph, Ph.D.

The Pennsylvania State University, 1994

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

The Pennsylvania State University

The Graduate School

THE EFFECTS OF STUDENT APTITUDE AND PRESENTATION STRATEGIES ON COMPUTER ANXIETY AND STUDENT ACHIEVEMENT OF COMPUTER SKILLS

A Thesis in

Instructional Systems

by

Thomas J. Leso

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

August 1994

We approve the thesis of Thomas J. Leso.

Date of Signature

05/26/94

Randal D. Carlson
Assistant Professor of Education
Thesis Advisor
Chair of Committee

5/27/94

Barbara L. Grabowski
Associate Professor of Education

6/3/94

B. Allan Quigley
Associate Professor of Education

Edmond Marks

6/1/94

Edmond Marks
Senior Research Associate
Affiliate Associate Professor of
Educational Psychology

6/3/94

Donald W. Johnson
Associate Professor of Education
In charge of Graduate Programs
in Instructional Systems

ABSTRACT

The objective of this research was to investigate the relative effectiveness of three presentation strategies (verbal, visual, and hands-on) and their effects on student achievement of computer skills necessary for computer programming and on computer anxiety. Another intention of this study was to ascertain whether computer science aptitude had an effect on student achievement of computer skills and on computer anxiety or interacted with presentation strategies.

Data were collected from 81 freshman and sophomore college engineering students who registered for an introductory technical FORTRAN programming course. Students were classified as having high computer science aptitude or low computer science aptitude based on test results. Three different presentation strategies were used to instruct students about computer skills necessary to program a computer. Two passive methods, verbal and visual, and one active method, hands-on, comprised the treatments. The criterion measures consisted of a written achievement test, a practical achievement test, and a computer anxiety questionnaire. Multivariate analysis of variance was conducted for the three-by-two factorial design.

Results of the study indicated that different instructional strategies for the training of computer skills have very little effect on computer anxiety over a one week investigative period. The hands-on presentation strategy

was superior to the visual presentation strategy in facilitating student achievement for the use of computer skills when measured by a practical test. Computer science aptitude was not found to be a significant factor in the study. These results suggest that different presentation strategies are not equally effective in facilitating student achievement of computer skills. The hands-on presentation strategy provided for a significantly more effective training method than did the visual presentation strategy. The hands-on training made available the cues that are later to be used in applying the computer skills. Other presentation strategies that do not allow for hands-on training may have omitted crucial cues that allow for effective learning strategies.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

THE NEED FOR THE STUDY

1.1. Introduction

The use of computer skills in education and society has become very prevalent during the information revolution. The need for computer skills is evident in most career job requirements and in the use of instructional technology. Word-processing, programming, electronic mail and computer-aided instruction are just a few of the applications that demand computer skills. To provide training for these skills, specific forms of instruction at K-12, college and in industry have been created to satisfy the need. Three common types of training of computer skills are lecture, demonstration, and self-paced tutorial (Harrington, 1990). The preferred choice among larger educational and training institutions has been the in-class lecture, a verbal presentation strategy (Bikson and Gutek, 1983).

A verbal presentation strategy is categorized as a covert or passive learning environment for students that employs the use of words by the presenter as mediators (Hortin and Bailey, 1982). Unfortunately, a problem with the verbal presentation strategy for learning computer skills is the lack of active, hands-on use of the computer by students in a computer classroom or a computer laboratory. This problem for learning computer skills can be investigated by the use of alternate presentation strategies.

One alternative is a visual presentation strategy that uses a large screen display as an additional mediator for students learning in a passive environment. The instructor demonstrates the computer skills at a computer located in the front of the classroom while students observe the visual output that is projected onto a large screen. The visual presentation strategy is particularly attractive among college administrators because it is a low-cost training method. One computer is used by the instructor vis-à-vis a classroom of computers used by the students.

Another alternative would encompass an active rather than a passive learning environment. The active learning environment would permit students to train on computers in a computer classroom or laboratory. This would represent a hands-on presentation strategy.

Another problem for learning computer skills is a severe impediment to novice computer students who fear the use of computers called "computer anxiety" (Harrington, 1990). Computer anxiety has also been labeled computer-phobia, cyber-phobia, and technostress (Paul, 1982). Paul estimated that 30% of American workers fear computers. Most learners possess some form of anxiety, which may be advantageous to learning, but too much anxiety in some cases and too little anxiety in others can deter learning (Spielberger, 1966). The information revolution has not been universally accepted by students expected to use computers. Their reluctance to use computers may emanate from a lack of computer knowledge or a more generalized fear of computers and technology. A lack of computer knowledge can be influenced by training, but computer anxiety offers a more challenging problem.

The study of computer anxiety and its effect on the learning of computer skills has produced mixed results. Some researchers (Howard, 1986; Simonson, 1987; Spielberger, 1983) attempted to explain computer anxiety via personality factors while other researchers (Harrington, McElroy, and Morrow, 1990) suggested that computer anxious people might very well attribute their anxiousness to environmental factors, particularly those involving specific features of training programs. Howard (1986) speaks of two dimensions of anxiety: duration (temporary versus permanent) and intensity (normal or neurotic). Simonson (1987) and Spielberger (1983) refer to two types of anxiety: trait (relatively stable individual differences) and state (stress-producing situations that vary in intensity or duration).

To date, research has concentrated on the development of measures of computer anxiety (Cattell, 1961; Howard, 1986; Simonson, 1987; and Spielberger 1983) and the search for correlates and possible causes of computer anxiety (Cambre, 1984; Howard, 1986; Morrow, 1986; Chen, 1986; and Cushall, 1989). Most studies have focused on the identification and measurement of individual differences such as math anxiety, math proficiency, computer science aptitude, gender, age, previous experience, or attitudes toward computers (Konvalina, Wileman, and Stephens, 1983; Jay and Willis, 1992). Very little research is available on the treatment of computer anxiety.

The results of Harrington's study (1990) suggest that for certain individuals computer anxiety can be influenced by the type of training provided and the receipt of a desired training approach. Such findings will help to support the concept that computer anxiety may be a temporary, normal form of state anxiety rather than a more permanent trait anxiety.

Many training efforts have as a goal to expose and train people to use computers with minimal personal discomfort and maximum proficiency. Harrington's findings have implications concerning interventions for the reduction of computer anxiety and achievement of computer skills. The fact that receipt of preferred mode of training (lecture vs. self-paced tutorial) affected post-experimental computer anxiety suggests that further consideration be given to training methods, in addition to individual difference variables such as previous experience, knowledge of computers, or aptitude for computer science.

Konvalina, Wileman, and Stephens (1983) have used computer science aptitude as a predictor for advising and placing students in different types of introductory computer science courses. A predictor such as computer science aptitude was used to place students into a less technical course than an introductory programming course to allow for a higher probability of success, increase computer knowledge with less stress, less withdrawers from the course, and better utilization of limited computer resources.

Banks and Havice (1989) report that computer anxiety can be successfully diminished if the initial training method focus on mastering computer skills before asking the student to master subject skills. That is, computer skills are a prerequisite for computer problem solving and computer programming. Banks and Havice further report that the instruction of computer skills should be realistic and should follow a careful step-by-step set of procedures, and should provide for lots of early success at the computer.

An associated problem of learning computer skills in a passive learning environment involves the testing of these skills. Students usually have their achievement of computer skills evaluated by a measure of knowledge associated with "what skills to use" rather than "performance on the computer." An instrument that measures "what skills to use," such as facts, concepts, and rules is usually in the form of a written test. The achievement of computer skills should be evaluated by a student's "performance on the computer." A practical test on the computer would be a form of evaluation commensurate with measuring computer skills.

The practical test is congruent with the learning objectives.

1.2. Problem Summary

The instructional effects of three types of presentation strategies (verbal, visual, and hands-on) will be investigated with respect to computer anxiety and facilitating student achievement of computer skills necessary for computer programming.

1.3. Problem Identification

Some questions that will come under investigation follow.

1. What is the entry level computer anxiety that is exhibited by engineering students required to take a computer programming course?

2. What aptitude for computer science do engineering students possess before taking a FORTRAN programming course?

3. Which presentation strategy for computer skills is the most effective training method for students with high and low aptitude for computer science?

4. Do any of the presentation strategies reduce computer anxiety when investigated by levels of aptitude for computer science?

5. Which presentation strategies are more effective for different types of learning outcomes by aptitude levels,

i.e., recognizing "what" computer skills to use, vis-à-vis, "performance of" computer skills at the computer?

6. Are there interactions between computer science aptitude and presentation strategies on achievement of computer skills and reduction of computer anxiety?

7. What are the differences in pre-treatment and post-treatment computer anxiety with respect to presentation strategies and levels of aptitude for computer science?

1.4. Research Questions

The following are research questions under investigation.

1. Are three types of presentation strategies (verbal, visual, and hands-on) equally effective in facilitating student achievement of computer skills?
2. Are the three types of presentation strategies equally effective in reducing computer anxiety?
3. Does computer science aptitude affect student achievement of computer skills and computer anxiety?
4. Is there an aptitude-by-treatment interaction between computer science aptitude and presentation strategies on student achievement of computer skills?

1.5 Hypotheses

The following hypotheses are presented as both null hypotheses (Hx.0)and research hypotheses (Hx.1, .2, ...).

They will be stated in terms of 5 main effects and an interaction effect. The 5 main effects are:

H1. Presentation strategy on written test;

H2. Presentation strategy on post-treatment computer anxiety;

H3. Computer science aptitude on written test;

H4. Computer science aptitude on post-treatment computer anxiety;

H5m. Presentation strategy on practical test.

The interaction effect is:

H5i. Presentation strategy with computer science aptitude on practical test.

H1.0 There will be no significant differences in mean written achievement test scores among three presentation strategies.

H1.1 Mean written achievement test scores for hands-on training are higher than mean written achievement test scores for visual training.

H1.2 Mean written achievement test scores for visual training are higher than mean written achievement test scores for verbal training.

H2.0 There will be no significant differences in mean post-treatment computer anxiety scores among three presentation strategies.

H2.1 Mean post-treatment computer anxiety scores for the hands-on presentation strategy are lower than mean post-treatment computer anxiety scores for visual and verbal presentation strategies.

H2.2 Visual and verbal presentation strategies have a common post-treatment computer anxiety mean.

H3.0 There will be no significant differences in mean written test scores of students with two levels of computer science aptitude.

H3.1 The mean written test score of students with high computer science aptitude is higher than the mean written test score of students with low computer science aptitude.

H4.0 There will be no significant differences in mean post-treatment computer anxiety scores of students with two levels of computer science aptitude.

H4.1 The mean post-treatment computer anxiety score of students with high computer science aptitude is lower than the mean post-treatment computer anxiety score of students with low computer science aptitude.

H5i.0 There will be no two-way interaction between presentation strategy and computer science aptitude on the practical test.

H5i.1 College students with high and low computer science aptitude and trained by hands-on presentation strategy have a higher mean practical test score than any other combination of the two factors.

H5i.2 The mean practical test scores for hands-on-high and hands-on-low groups are equal.

H5i.3 The mean practical test scores for the visual-high and verbal-high groups are higher than the mean practical test scores for the visual-low and verbal-low groups.

H5m.0 There will be no significant differences in mean practical test scores among three presentation strategies.

H5m.4 Mean practical test scores for the hands-on presentation strategy are higher than mean practical test scores for the visual presentation strategy.

H5m.5 Mean practical test scores for the hands-on presentation strategy are higher than mean practical test scores for the verbal presentation strategy.

1.6. Project Significance

This research project proposes to provide justification for the use of specific presentation strategies that promote the effective learning of computer skills and reduce computer anxiety. The intended outcomes include:

- determination of effective presentation strategies...higher achievement levels with superior training methods.
- reduction of computer anxiety...less anxious students with better training methods.

1.7. Rationale for Significance

Traditional methods of instruction such as lecturing and reading of printed materials may not provide for effective presentation strategies when computer skills are involved in the learning process. Hands-on training provides a realistic dimension for the instruction of computer skills. Step-by-step tasks that can be performed successfully by the learner provide for a positive computer experience and a reduction to computer anxiety (Leso and Peck, 1992). Furthermore, hands-on use of the computer to evaluate achievement of computer skills is congruent with the specific educational objectives.

1.8. Generalizability of the Study

The factors that affect the generalizability of this study or external validity of the experiment will be discussed in terms of the sample population, the programming course, and computer anxiety.

Population validity is the extent to which one can generalize from the experimental sample to a defined population. In this study, the experimentally accessible population consisted of freshman and sophomore engineering students from a Penn State commonwealth campus. The engineering students were predominately white, male subjects that came from mainly eastern United States.

The technical, programming course that was used for this experiment was an introductory computer programming course for engineers. The course included development and implementation of algorithms in a procedure-oriented language, with emphasis on numerical methods for engineering problems. The 3-credit, general quantification skills course is required by all engineering majors enrolled in a baccalaureate degree program of the college of engineering. During the 3rd week of the course, the computer science aptitude test was administered as a usual classroom event to evaluate problem solving and algorithmic skills. The written and practical achievement tests occurred at plausible periods in the course calendar and were embedded in weeks 4 and 5 of the course as regular examinations.

Pre-treatment anxiety levels of subjects were recorded by an anxiety inventory questionnaire (Spielberger, 1983). The reported college-level normative sample, describing students enrolled in introductory psychology courses at the University of South Florida, had mean = 37.6 and sd = 11. A Penn State commonwealth campus study (Leso and Peck, 1992), using the same questionnaire and describing students enrolled in (a) programming course reported mean = 35.7 and sd= 14.8; and (b) non-programming course reported mean = 35.8 and sd = 11. Both of these studies reported anxiety levels comparable to those exhibited by subjects in this study. The pre-treatment anxiety levels for subjects in the verbal treatment group (mean = 39.7, sd = 10.6), in the visual treatment group (mean = 40.3, sd = 15), and in the hands-on treatment group (mean = 40.5, sd = 10) were common.

Chapter 2

LITERATURE REVIEW

The literature review contained in this chapter focuses on critical variables in the design of this study. First the reader will be introduced to an effect that places stress on human performance under investigation--computer anxiety. Next, the teaching and learning process will be reviewed with respect to learning strategies, varieties of learning, and presentation strategies. Finally, realism is emphasized in the context of different presentation strategies which will be used for the training of computer skills under investigation.

2.1. Computer Anxiety

Computer anxiety is defined as the fear or apprehension felt by individuals when they use computers or when they consider the possibility of computer utilization (Simonson, Matt and Maurer, 1987). Researchers have reported that anxiety is the result of psychological stress and that the concept of anxiety consists of two types: trait and state anxiety (Cattell, 1966; Cattell and Scheier, 1961, 1963). An operational definition of trait and state anxiety has been elaborated by others (Spielberger, 1966, 1972, 1976, 1979; Jonassen, 1986). Trait anxiety refers to relatively stable individual differences in anxiety-proneness, that is, differences between people in the tendency to perceive stressful situations as dangerous or threatening and to respond to such situations with elevations in the intensity of their state anxiety reactions. State anxiety is situational; it results directly from some stress-producing situation during a finite period of time. The stronger the anxiety trait, the more probable that the individual will experience more intense elevations in state anxiety in a threatening situation. Computer anxiety is generally regarded as an instance of state anxiety and is susceptible to changes over time (Cambre and Cook, 1984).

Researchers have developed tests and other instruments to measure computer anxiety and have investigated possible correlates of computer anxiety. One such instrument was derived from an index to measure intentions to use computers among people entering the teaching profession by Simonson (1987). Maurer (1987) later modified this questionnaire by omitting references to the teaching profession in the wording scale and by adding items to capture feelings of anxiety, in addition to intentions, about using computers. This measure, which was based on an unpublished thesis, was called the computer anxiety index (CAIN). It used a twenty-six item, six-point Likert computer anxiety index to measure the trait of computer anxiety and to be predictive of the development of the state of computer anxiety. Primarily it measured: (1) the avoidance of computers and areas where computers are located; (2) excessive caution when using computers; (3) negative remarks toward computing and persons using computers; and (4) attempts to shorten periods when computers were being used (Simonson, Matt, and Maurer, 1987).

Another instrument that has been used by researchers to measure anxiety was the State-Trait Anxiety Inventory (STAI) self-evaluation questionnaire developed by Spielberger (1979). The STAI has been used as a pretest, post-test, and midterm test to measure computer anxiety in educators learning to use the computers (Spielberger, 1983; Honeyman and White, 1987). The concepts of state and trait anxiety were first introduced by Cattell (1961, 1966) and have been elaborated by Spielberger (1966, 1972, 1976, 1979). Trait anxiety refers to relatively stable individual differences in anxiety-proneness while state anxiety refers to a process taking place at a given time and level of intensity. Spielberger suggested that whether or not people who differ in trait anxiety will show corresponding differences in state anxiety depends on the extent to which each of them perceives a specific situation as psychologically dangerous or threatening, and this is greatly influenced by each individual's past experience.

The STAI has been used extensively in research and clinical practice. It is comprised of separate self-reporting scales for measuring state and trait anxiety. The state anxiety scale consists of twenty statements that evaluate how respondents feel "right now, at this moment." The trait anxiety scale consists of twenty statements that assess how people generally feel. Consistent with the definition of state anxiety given above, the essential qualities evaluated by the STAI state anxiety scale are feelings of apprehension, tension, nervousness, and worry. In addition to assessing how people feel "right now," the state anxiety scale may also be used to evaluate how they felt at a particular time in the recent past and how they anticipate they will feel either in a specific situation that is likely to be encountered in the future or in a variety of hypothetical situations. Results of the use of the state anxiety scale by researchers Honeyman and White (1987) have indicated that anxiety levels had changed significantly after thirty hours of contact with the computer. The state anxiety scale measured a lower mean score on the post-test evaluation given after a pretest state anxiety measurement was taken before the thirty hours of hands-on use by teachers with computer software packages such as word processing and spreadsheets.

Correlates of computer anxiety have been explored along with measurements of anxiety indices and scales with CAIN and STAI. The factors included in the explorations of correlates of computer anxiety have been locus of control, cognitive style, rigidity, math anxiety, trait anxiety, computer knowledge, computer experience, grade point average, gender and age (Raub, 1981, Howard, Murphy, and Thomas, 1987; Morrow, 1986; Chen, 1986; McCormick and Ross, 1990; Marcoulides, 1988). Results of these studies indicated that self-assessments of computer knowledge and computer experience explain more of the variance in computer anxiety than do personality/attitudinal correlates (e.g., locus of control, rigidity, math anxiety). This implies that computer anxiety may be more a function of prior computer experience, a modifiable condition, than a deeply entrenched attitude or personality trait. That is, the more successful the computer experience, the less the computer anxiety. It also suggests that the definition of computer anxiety may align with Spielberger's definition of state anxiety (that is situational--resulting directly from some stress-producing situation during a finite period of time) and that computer anxiety may be greatly influenced by an individual's past experiences. Marcoulides (1988), McCormick, and Ross (1990) reported that performance of and achievement of computer skills seem to be influenced by computer anxiety and past computer experiences. However, it is still not clear whether computer anxiety is strictly a trait or a state. Additional research is necessary to measure, test, and evaluate computer anxiety and determine whether computer anxiety is a situational or more enduring form of anxiety.

Amid controversy about the reality of traits and measurements of computer anxiety, this study takes the view suggested by Cambre and Cook (1984) and the findings of Harrington, McElroy, and Morrow (1990) that support the conceptualization of computer anxiety as a state rather than as a more permanent trait. In past research efforts, Simonson and others (1987) have used a Computer Anxiety Index (CAIN) to measure the trait of computer anxiety and to be predictive of the development of the state of computer anxiety. Spielberger and others (1983) have used the State-Trait Anxiety Inventory (STAI) questionnaire under the assumption that if an individual had a trait of computer anxiety, he/she would develop a state of anxiety while seated before a computer, and this state could be measured by STAI. If computer anxiety can be measured reliably and can be changed over time, then the most effective method of overcoming computer anxiety may be to confront the problem directly by computer education. The preliminary efforts by Harrington and others (1990) to understand whether training methods can alleviate computer anxiety suggested that tailoring training to meet individuals needs may be helpful in overcoming computer anxiety.

All research studies of computer anxiety referenced herein have reported a negative correlation between computer anxiety and prior experience with computers. One such study was conducted with a group of undergraduate students enrolled in the same introductory computer class (Jones and Wall, 1985). Using the CAIN, pretest and post-test anxiety, gender, age, and computer experience were evaluated. The relationship between experience and post-test anxiety scores was significant. Students that claimed to have prior computer experience recorded low levels of anxiety after the use of a word processing package. Using the STAI, another study involving adults in a computer literacy class found age and gender to have no significant correlation with computer anxiety (Honeyman and White, 1987). Using the Computer Anxiety Scale (CAS), a third study reported that computer anxiety seemed to be a good predictor of computer achievement in college students taking computer classes (Marcoulides, 1988). Marcoulides reported that the more computer experience the students had, the less computer anxiety was recorded. He also reported that computer achievement was more related to computer anxiety ($r=-0.71$) than to computer experience ($r=0.37$).

Further review of studies has indicated that computer anxiety was reduced with hands-on experience with computers (Cushall, 1989). A pretest post-test experiment involving an introductory computer course found very significant decreases in computer anxiety over the term of the course (Howard, Murphy, and Thomas, 1987). The course was composed of BASIC language concepts and programming assignments and VISICALC. BASIC, a programming language, is considered somewhat unfriendly, while VISICALC, a spreadsheet software application, is generally regarded to be user-friendly. The order of presenting the programming language and the spreadsheet software was reversed between two sections of an introductory, undergraduate course but there was no significant difference between the two experimental groups (BASIC-1st, VISICALC-2nd, or VISICALC-1st, BASIC-2nd) when using the hands-on training method. Howard also found that 33 percent of undergraduates without previous computer experience had seriously high levels of computer anxiety. He suggested that it may be advantageous to segregate initially anxious students into separate classes of instruction.

Ongoing research is investigating the relationship between changes in the level of computer anxiety and the type of computer courses offering introductory hands-on computer experience (Leso and Peck, 1992). For example, is there more computer anxiety associated with a programming course than with a non-programming course that offers tools software applications such as spreadsheets, data bases, and word processing? Could it be that programming a computer is more stressful to individuals who possess little or no computer experience or computer science aptitude than simply using pre-programmed tools software applications? If we are to confront the problem of computer anxiety directly by offering computer experience through introductory computer courses, then we must choose the best type of teaching methods that are most effective in reducing computer anxiety at all levels of computer skills.

Harrington and others (1990) report that computer anxiety can be influenced by the type of instructional method and the receipt of a desired training approach. They suggest that the goal of further research should not be to debate computer anxiety, but to train people to use computers with minimal personal discomfort and maximum proficiency. This finding and another (Cambre and Cook, 1984) help to support the conceptualization of computer anxiety as a state and subject to change rather than as a more permanent trait. Another researcher, Howard (1986), approached anxiety from two slightly different dimensions: duration (temporary versus permanent) and intensity (normal or neurotic). This type of research pursued multiple dimensions of anxiety and tried to determine what form computer anxiety best approximated. Such research is important from a training perspective because it addresses the behavior and individual needs of the learner. If computer anxiety is a temporary, normal type of anxiety, then successful training on computers may be sufficient to make an individual computer functional. Howard (1987) suggested that tailoring training to meet individual needs such as segregating initially anxious students into separate classes may be helpful in achieving the goal of minimizing computer anxiety and maximizing achievement.

Teaching methods and expectations for computer use were evaluated by Banks and Havice (1989). This study reported that computer anxiety can be successfully diminished if the instruction takes into consideration that initial learning must focus on the (prerequisite programming) skill itself, then gradually give way to applications of the skill to subject specific instructional goals (like programming). The initial learning should provide for a "safe learning environment," says Banks and Havice, that is, allow the student to master computer skills before asking the student to master subject skills. Careful step-by-step instruction should be provided with lots of early success built-in to help diminish computer anxiety. The instruction should be realistic and expect things to go wrong now and then. The content of instruction of computer assignments should be simple at first, to focus on the more important skill of learning the use of the computer. Banks and Havice as well as other researchers such as Howard, Cambre, Cook, and Harrington who have investigated computer anxiety and training on microcomputers stress the need for lots of successful experiences (immediate hands-on computer experiences) to reduce the fears of using computers.

2.2. Learning Strategies

Learning strategies can be defined as behaviors and thoughts in which a learner engages and which are intended to influence the learner's encoding process (Weinstein and Mayer, 1983). Learning strategies are one part of the framework for describing the teaching-learning process. Another part of the framework is teaching or presentation strategies which will be addressed in section 2.4. Other elements of the teaching-learning process are teacher and learner characteristics, encoding (including internal cognitive processes), learning outcomes (that depend on both presentation and learning strategies), and performance (including achievement on tests). The use of particular learning strategies during learning can affect the encoding process, which in turn affects the learning outcome and performance. The goal of any particular learning strategy may be to affect the way in which the learner selects, acquires, organizes, or integrates new knowledge.

Some major categories of learning strategies (identified by Weinstein and Mayer, 1983) are: (1) rehearsal strategies such as copying, underlining, or shadowing of texts; (2) elaboration strategies such as paraphrasing, creating an analogy, forming a mental image, or summarizing lectures; (3) organizational strategies such as outlining, creating a sequence of steps, grouping or ordering commands to be learned from a list, or creating a hierarchy of tasks; (4) comprehension monitoring strategies such as checking for comprehension failures through self-questioning of material presented in class and using questions from exercises related to the text; and (5) affective strategies such as being alert and relaxed, to help overcome test anxiety, or anxiety reduction techniques which involve the effective use of available resources.

Weinstein and Mayer pointed out that learning strategies are general techniques which are just a part of the arsenal of knowledge that a learner needs for effective learning. Learning is also enhanced when the learner possesses a great deal of domain-specific knowledge. Simon (1980) summarizes this point as follows: "The scissors does indeed have two blades and ... effective ... education calls for attention to both subject-matter knowledge and general skills" (p.86). Cronbach and Snow (1977) emphasized this point too, particularly in teaching of a skill, where the criterion of excellence is well defined (i.e., performance is either successful or unsuccessful). Teaching (computer) skills require considerably more than turning the learner loose to practice (with a computer use's guide). Cronbach suggested task analysis (i.e., subdividing major tasks into smaller, more manageable events that the expert performers use) to pave the way for organizing lessons and the use of realism (i.e., realistically produce instructional cues that affect direct significant training action). He postulated the first rule in training for a skill, that is, make available the cues that are later to be used in applying the skill. This may seem obvious, but all too often teaching or presentations strategies have omitted crucial cues. For example, the anxious computer user gets none of the feel of the point, click, and drag operations of a computer mouse while reading about the use of these steps in the text or observing the instructor do it at a computer in the front of the classroom while the screen output is projected onto a large screen display

Learning strategies may be triggered by specific instructional methods to elicit a particular type of performance objective. For example, sight and hearing are considered to be the primary senses for information acquisition that are used during a lecture. This type of instruction is represented by a verbal and textual presentation strategy (Rickards, 1980; Rickards and Divesta, 1974). A more visual presentation strategy adds additional cues (such as pictures or images that represent a more realistic stimulus) to assist the learner in organizing information (Dwyer, 1979). These two methods of instruction are considered to have passive student participation since direct student interaction is at a minimum with the learning materials. A third method of instruction is a practical, hands-on approach for students who physically and actively interact with the learning materials (Dale, 1954, and Finn, 1953). This presentation strategy employs a realistic stimulus and actively engages the student in the learning process.

2.3. Varieties of Learning

Methods employed by instructors have two distinctly different kinds of goals in the classroom. The goals are: (1) teaching students "what" to learn; and (2) teaching students "how" to learn. Gagne's (1977) answer to the question, (what is to be learned?) suggests five different learning outcomes are possible: intellectual skills, verbal information, motor skills, cognitive strategies, and attitudes. Fitts and Posner's (1967) answer to the question, (how to learn skills?) suggests a hierarchy of habits that result in a relatively permanent change in performance that can be shown to be the result of experience. Instruction in the use of computer skills requires attention to both of these goals.

According to Gagne (1977), eight kinds of learning can be distinguished: signal learning, stimulus-response learning, chaining, verbal association, discrimination, concept, rule, and problem solving. Problem solving learning is the most complicated. It involves discerning a new rule that combines rules previously learned. Each step rests on the previous activity, and the experience of each ends with a different capacity for performance. Computer skills require step-by-step success with the use of discriminate, interactive commands with the computer that follow a hierarchical set of rules and sequences to accomplish desired tasks. Computer programming involves the collective use of computer skills and problem solving learning.

The concept of hierarchical and sequential organization is basic to an understanding of the processes involved in skilled learning and performance. Fitts and Posner (1967) stated that the concept of hierarchy implies two operations: (1) it must be possible to group events, symbols, functions, etc., into categories; and (2) it must be possible to specify some restrictive relationship of position, order, or sequence between categories. For examples, hierarchical organizations may be formal systems of logic and mathematics, or natural languages with their grammars, as well as motor sequences, like sentences that must follow a restricted course. Learning skills involve a new integration and ordering of units that begins even before birth and continues with acquisitions of new skills through adulthood. There are three phases that appear to be involved in the acquisition of complex skills. They are early or cognitive phase, intermediate or associative phase, and final or autonomous phase. During the last phase, component processes become increasingly autonomous, less directly subject to cognitive control, and less subject to interference from other ongoing activities or environmental distractions. However, in the early or cognitive phase, environmental distractions may present serious impediments to learning.

One such distraction by a specification of the demands that the (computing) environment places on the individual is stress (computer anxiety). When stress is defined by the demands a task makes, it is immediately apparent that people do their best under immediate conditions of stress, said Fitts and Posner. Remove all input--all environmental variation, all demands--and the individual at best becomes bored, loses alertness, and perhaps goes to sleep. People also do poorly at the other extreme of stress. Increase the task load to a point where it is impossible for the individual to keep up with the demands placed upon him/her, or change the environmental conditions until they approach the limits of tolerance for stress, and performance again deteriorates. Moreover, the effects of stress are not static, but change as the task continues. In this study, the investigation of different presentation strategies will focus on effective teaching styles that consider environmental distractions with emphasis on their impact on computer anxiety and achievement of computer skills.

Some of the more useful research on learning styles (i.e., a learner's consistent way of responding in learning situations) has been on cognitive styles, that is, how a learner perceives, thinks, remembers, and solves problems in a consistent, identifiable way. Cognitive styles serve as useful devices to distinguish student attitudes, temperaments, and motivations in a variety of situations. How students feel about computers (their level of computer anxiety) may affect the situation. Wittrock and Lumsdaine (1977) assert that learning is more effective when the educational method (say, hands-on instruction) and the learner's ability (say, low computer science aptitude) are congruent. Learners have different needs, motivations, and interests plus learning styles. To make instruction more effective, O'Neil and Spielberger (1979) suggest that instructors must know more about these characteristics in the students. To evaluate students' computer anxiety and aptitude levels before instruction may benefit as predictors for the types of instruction.

McKeachie (1980) has said that the effectiveness of student learning depends to some extent upon the strategy used by the students. Cronbach and Snow (1977) have pursued this notion with their theory of aptitude-treatment interactions, which study the differences in students' aptitudes for learning under different types of instructional methods. It means that individual differences in personality will help to predict learning outcomes.

Aptitude-by-Treatment Interaction (ATI) in this study will examine the individual difference variable of aptitude in computer science with the types of instructional presentations as joint determiners of academic achievement and the relationships with computer anxiety.

Aptitude can be defined as the ability to learn quickly and easily. It is directly related to what an individual learns, how he/she learns it (learning style), how much he/she learns, and how rapidly he/she acquires content information from a specific instructional presentation. Different types of instructional presentations may be programmed instruction, television, computer aided instruction, large lecture format, verbal, visual, and hands-ons. Researchers such as Cronbach (1977), Gagne (1977), and Dwyer (1978) have suggested that different students will profit differently from identical presentations and that there is no single best instructional process or format which will produce optimal learning for all students. They contend that students learn from instructional presentations to the extent that their aptitudes enable them to interact with the learning materials. They further suggested that students should be assigned to treatments identified as being most beneficial for a particular configuration of aptitudes.

2.4. Presentation Strategies

With respect to the three types of presentation strategies mentioned previously (namely, verbal, visual, and hands-on), computer hands-on learning methods have been sparsely investigated in the research literature. However, verbal and visual learning methods have been evaluated at varied depths of information processing on undergraduate students' memory systems in terms of their ability to acquire and retrieve information related to different types of educational objectives.

Paivio (1971) postulated the dual-code theory in which there are two separate but interconnected memory systems; one is verbal and the other is visual. This theory suggested that information is stored verbally in the verbal system and information is stored visually in the visual system. Its representation resembled its input. Investigations by Levie, W. and Levie, D. (1975) provided support for the dual-code theory. They said that pictures are remembered better than words. However, they suggested that the two systems interact. These investigations questioned whether different strategies are more effective in facilitating achievement, whether certain strategies require learners to spend more time interacting with instructional content, and whether verbal or visual testing requires students to spend more time processing and responding to retrieval cues.

For verbal learning methods, results indicate that organization and elaboration of textual data improves the acquisition and retrieval of declarative knowledge (Glynn and DiVesta, 1977; Wittrock, 1977). Organization is the process of dividing an information set into subsets and indicating the relationship among subsets. Elaboration is the process of adding to the information being learned (Gagne, E., 1985). For visual learning methods, results indicate that different rehearsal strategies used to complement visualized instruction are not equally effective in improving student achievement (Dwyer, 1985). Some specific type of visual is effective for some specific objective, but all types of visuals are not equally effective. Dwyer (1979) and Paivio (1971) both reported that visuals did improve learning of prose materials.

Verbal and visual presentation strategies have been advocated by Hortin and Bailey (1982) to improve classroom instruction. They have identified these strategies through their research of mental rehearsal techniques. Mental rehearsal was divided into two interrelated processes called verbal rehearsal and visual rehearsal. Verbal rehearsal is defined as information processing which requires the use of words or text as mediators. Visual rehearsal is defined an information processing using icons (such as pictures, computer output projections on large screen displays, drawings, or diagrams) as mediators.

Hands-on versus demonstration presentation strategies were investigated by Henry and Holtan (1987). Two groups of preservice teachers were first presented the same one-hour lecture/demonstration which provided an overview of microcomputer operations, terminology, and classroom applications. Next, the hands-on group worked through individual packets of material which included reviewing several examples of CAI software to acquire hands-on laboratory computer experience. The other group of preservice teachers was given an in-class demonstration by an instructor and a discussion using the same materials. The results concluded that there was no significant difference in the amount of change in knowledge between the two groups, but there was a significant positive change with respect to attitude toward the instructional use of microcomputers by the demonstration-only group. Henry and Holtan suggested that it might be better to begin training with an interesting discussion and demonstration rather than trying to schedule immediate individual microcomputer hands-on work.

Hands-on presentation strategies have been recommended by Banks and Havice (1989). They suggest that instruction should be realistic when training computer skills. Hands-on rehearsal is defined as information processing which requires the active use of the computer by the student. The interaction of the student with the learning object acts as the mediator of the hands-on presentation strategy.

Most research studies investigate the following independent variables: performance criteria; learning sequence; learning strategies (what must one learn and what must learner do to learn); presentation strategies and media; motivation; and individual differences. Some researchers point to certain factors that are important for preparing instruction based on their research findings. Prior knowledge, I.Q., and oral reading comprehension are considered three of the most important factors by Dwyer (1978). This current study considers computer anxiety and computer science aptitude as important factors for preparing instruction of computer skills.

Methods of presenting instruction can be either self-paced or externally-paced instruction. Self-paced allows each student to progress at his/her own rate. Externally-paced instruction is presented to a group of students. Material is programmed linearly and presented sequentially. Students do not have control of the rate. All students learn the same content at the same rate and time. Presentation strategies in this study are all externally-paced by the instructor.

Studies evaluating externally-paced visualized instructions have the following implied assumptions: all students learn at the same rate and have the same material sophistication regardless of their different background. A good instructional design facilitates externally-paced visualized instruction. Objectives for externally-paced experimentation have been to investigate the realism continuum, find which type of visuals are more effective, and evaluate immediate and delayed retention. Findings from externally based investigations show that the realism continuum was not effective. Some cues are better than others. Males and females learn equally. Different I.Q. results in different effectiveness. An explanation as to why these studies contradict the realism theory is that realistic visuals present too many details to be interacted in a given time when individual differences are not considered, such as aptitude and anxiety. Students may perceive visuals as redundant. Students are not used to this kind of learning (Dwyer & Dwyer, 1987).

Investigations (by Dwyer, C., 1984, 1986; and by Dwyer, F., 1978, 1985) of the effects of passive and active learning strategies have produced varied results in the literature, partly because of the type of learning (e.g., movement vs. memory) and the presentation methods (e.g., self-paced vs. externally-paced) used to produce the rehearsal. Active rehearsal strategies or overt presentation strategies engage the learner in some form of outward or physical activity to aid in the encoding of information in long-term memory. Passive rehearsal or covert presentation strategy deals with internalizing or mentally possessing information. Mixed results have been reported by one such study that examined the effectiveness of observable visual imagery strategy for the encoding of verbal information with special attention given to the relative effectiveness of the strategy for field-dependent and field-independent learners. Field-independent subjects benefited most from simple line visuals that supplemented text and scored markedly lower when required to engage in the observable visual imagery strategy. In contrast, field-dependent subjects scored higher when required to engage in the observable visual imagery rehearsal (Joseph, 1987). The control group was presented text on the computer without visuals or rehearsal strategies. A study by Dwyer (1985) reported that a form of active rehearsal in which students shaded in portions of a drawing of the human heart resulted in an effective overt rehearsal strategy to facilitate learning.

Based on research of presentation strategies for verbal and visual instruction, the following is summarized. Some of the limitations associated with oral and printed instruction include: words fail to communicate precisely the intention and words are not easy to interpret; not all students get the same amount of information due to different backgrounds; and print does not have realistic details. The realism theorists suggest that learning will be more complete as the number of clues in the learning situation increase.

Some reasons for using visualization in instruction are: there are increases in learner interest, motivation, curiosity, and concentration; visualization spans linguistic barriers; it presents relationships of locations, parts, and overcomes time and distance; it can organize new material to be presented; and it can emphasize and reinforce oral and printed instruction.

2.5. Realism Theories (Dale, 1954)

The theoretical orientations of visualization are referred to collectively as realism theories. The basic assumption centers around the following: Dale's (1946) cone of experience; Morris (1946) iconicity theory; and Carpenter (1953) sign similarity orientation--which says, "Learning will be more complete as the number of cues in the learning situation increases," and "an increase in realism in the existing cues in a learning situation increases the probability that learning will be facilitated."

Finn (1973) and Dale have recommended that for instructional purposes "the more realistic or lifelike the stimulus is, the greater the probability it has for facilitating learning," (referred to as the realism continuum). Also, other researchers have substantiated the basic concepts presented by Finn and Dale: e.g., Gibson (1954), projective-conventional continuum; Osgood (1963), more detachable-less detachable continuum; and Knowlton (1964), transparency-opacity continuum. In this research the opinion of the investigator is that computer skills should be learned on the computer to provide a more realistic and practical experience.

2.6. Multiple Channel Communication Theory (Hartman, 1961)

Should computer skills be learned by seeing and doing at the computer? Does the learner function as a multiple or single information processing system? Multiple channel communication (Hartman, 1961) is generally considered to involve a continuous simultaneous presentation of information over two or more channels. Multiple channel communication is supported by realism theories of Dale's cone of experience which extends from the object or situation itself to a very simplified representation of the object or situation, i.e., concrete to abstract. The basic idea is that redundancy and more cues increase discriminations. There is repetition of content (usually sight and hearing are considered to be the primary senses for information acquisition) and consequently, the combination of oral and print will provide additional stimuli reinforcement to assist the learners in organizing the structuring of information. Carpenter's sign similarity hypothesis signals the use of symbols that possess a high degree of similarity to objects, (e.g., computer commands like copy, save). Morris' iconicity postulates a degree of similarity of the symbol to the object or situation, (e.g., computer icons). The sign similarity approach has shown an increase in learning when testing was done in the same situation in which the information was presented. Likewise, with cue summation, learning was shown to increase as the number of cues increase, (e.g., command, icon, key stroke).

However, there are limits to the amount of information a learner can process. Even though both theory and research suggest that the basic assumptions underlying the realism theories will increase learner acquisition of information, other researchers (Hsia, 1971; Dwyer, 1978) express concern about the limited capacity of the central nervous system and the auditory and visual information processing modalities. Single channel theorists suggest that when audio and visual/print are presented simultaneously, the video/audio channel suffers from stimuli competition. And less learning occurs when using bimodal presentation of information as a result of faltering in the central nervous system (Hsia, 1971). Overall, single and multiple channel communication support exists.

Chapter 3

DESIGN AND PROCEDURES

3.1. Overview

In order to test and evaluate the research hypotheses based on three presentation strategies (verbal, visual, and hands-on) for student achievement of computer skills and reduction of computer anxiety, a research design of six treatment groups was developed. Given the three types of presentation strategies and two levels of computer science aptitude (high and low), the experiment constitutes a 3-by-2 factorial design.

3.2. Design

Students that registered for three different sections of a Computer Science 201 course comprised the three groups who received the three types of presentation methods. Each of the three groups was further subdivided by two levels of computer science aptitude that was recorded on a computer science aptitude test taken during the third week of the semester. The duration of the experiment covered week three through week five of the 15 week semester dedicated to an introduction to computer programming, or a total of five 75-minute class/laboratory meeting periods for each presentation group. Periods one and two occurred on Tuesday and Thursday of week three. Periods three and four occurred on Tuesday and Thursday of week four. Period five occurred on Tuesday of week five. The major events for each period were the following:

- Period one was to test students in order to classify them by high or low computer science aptitude.
- Period two was to measure pre-treatment computer anxiety.
- Period three was to administer presentation strategies and written test.
- Period four was to accomplish laboratory exercises of computer skills.
- Period five was to take the practical test and measure post-treatment computer anxiety.

The computer skills achieved by the students involve pre-requisite knowledge for learning computer programming. That is, students are required to master computer skills before they can pursue the content area of programming a computer. The computer skills presented in the lesson consist mainly of the use of a programming language editor, compiler/interpreter, and debugger called WFOR (Waterloo microFORTRAN). WFOR is a software program that is used to create, save, get and change a FORTRAN program file. WFOR is dependent on the IBM hardware and VM/CMS operating system which are used to process and execute a FORTRAN program. VM/CMS commands compose the auxiliary computer skills presented in the lesson. The CMS commands are necessary to execute/run/debug a FORTRAN program file and to list-off the program file/output/directory to the printer.

Computer anxiety was measured with the State Anxiety Inventory (STAI) form developed by Spielberger (1983). A pre-treatment computer anxiety score was recorded from the STAI Form Y-1 self-evaluation questionnaire for state anxiety. The state anxiety questionnaire requested the students to indicate how they felt right now, at this moment, to a given statement about computers. The moment was chosen to coincide with a stress-producing situation as defined by Spielberger's reference to state anxiety. The exact moment the questionnaire was administered immediately preceeded the first use of the computer by the students in the course. This was recorded as the pre-treatment computer anxiety level.

The post-treatment computer anxiety level was recorded immediately following the practical test of computer skills performed at the computer. This moment coincided with the stress-producing situations of having to use the desired computer skills at the computer to demonstrate mastery of the learning outcomes. The STAI Form Y-1 self-evaluation questionnaire for state anxiety was used.

Achievement of computer skills was measured by two types of tests at two different times. A written test was administered to each computer science section immediately following the presentation in the classroom of the verbal and visual training methods, or (in the laboratory) of the hands-on training method. A practical test at the computer was administered to each computer science section in the computer laboratory one week after the written test was taken. Between the written test and the practical test each computer science section was given a set of exercises of computer skills to practice at the computer in the computer laboratory to permit all groups to reach criterion.

Figure 3.1 depicts the overall design of the study. Investigation of student achievement of computer skills that was measured immediately after the treatments with a written test merely determined "what" computer skills the learner remembered from the instructional presentation. Investigation of student achievement that was measured by the practical test on the computer determined if the learner could "perform" the desired computer skills after reaching criterion on the computer with practice of computer exercises in the laboratory.

3.3. Subjects

College freshmen and sophomore engineering students enrolled in Computer Science 201 at the Penn State Altoona Campus comprised the target population. Eighty-one students registered for the course prior to the Spring semester of 1992. Students scheduled one of the three sections of Computer Science 201 when course registration occurred. Twenty-seven students registered for section 1, section 2, and section 3, respectively.

COURSE REGISTRATION	COMPUTER SCIENCE APTITUDE TEST	PRE-TEST COMPUTER ANXIETY	PRESENTATION STRATEGY AND APTITUDE TREATMENT	IMMEDIATE WRITTEN ACHIEVEMENT TEST	COMPUTER LAB EXERCISES	DELAYED 1-WEEK PRACTICAL TEST	POST-TEST COMPUTER ANXIETY
CS 201 SECTION 1	LOW APTITUDE - - - - HIGH APTITUDE	MEASURE "FEELINGS" PRIOR TO COMPUTER USE	VERBAL LOW - - - - VERBAL HIGH	TEST "WHAT" COMPUTER SKILLS TO USE	REACH CRITERION W/ COMPUTER EXERCISES	TEST PERFORMANCE OF COMPUTER SKILLS	MEASURE "FEELINGS" AFTER COMPUTER USE
CS 201 SECTION 2	LOW APTITUDE - - - - HIGH APTITUDE		VISUAL LOW - - - - VISUAL HIGH				
CS 201 SECTION 3	LOW APTITUDE - - - - HIGH APTITUDE		HANDS-ON LOW - - - - HANDS-ON HIGH				
	45-MINS	30-MINS	45-MINS	30-MINS	75-MINS	30-MINS	30-MINS
	PERIOD 1	PERIOD 2	PERIOD 3		PERIOD 4	PERIOD 5	
	TUES	THURS	TUES		THURS	TUES	
BEFORE SEMESTER	SEMESTER WEEK 3		SEMESTER WEEK 4			SEMESTER WEEK 5	
TIME==>							

LEGEND: 3-by-2 FACTORIAL DESIGN WITH INDEPENDENT VARIABLES--
3 PRESENTATION STRATEGIES (VERBAL, VISUAL, HANDS-ON)
(CS 201 SECTION 1 = VERBAL; CS 201 SECTION 2 = VISUAL;
CS 201 SECTION 3 = HANDS-ON)
2 COMPUTER SCIENCE APTITUDES (LOW, HIGH).

AND WITH DEPENDENT VARIABLES--
IMMEDIATELY AFTER TREATMENT WRITTEN ACHIEVEMENT TEST
DELAYED 1-WEEK PRACTICAL TEST
POST-PRACTICAL TEST COMPUTER ANXIETY.

FIGURE 3.1. Overall Design of Study

Section 1 was designated as the group to receive the verbal presentation strategy. Section 2 was designated as the group to receive the visual presentation strategy. Section 3 was designated as the group to receive the hands-on presentation strategy. Each student within each group was classified as high computer science aptitude or low computer science aptitude based on the results of the Computer Science Aptitude Test taken during the third week of the semester.

3.4. Materials

The materials for this study consisted of an instrument to classify computer science aptitude among undergraduate engineers, an instructional script that formed the basis of the treatments of presentation strategies, laboratory exercises of computer skills to reach criterion, and criterion measures.

3.4.1. Computer Science Aptitude Test

The computer science aptitude test (Konvalina, Stephens, and Wileman, 1981) was used to decide if learners have a high likelihood of success in computer courses. The aptitude test (see Appendix B) has 4 parts or a total of 25 multiple choice questions. Questions 1 through 10 cover sequences and logic. Questions 11 through 15 simulate a calculator. Questions 16 through 20 trace an algorithm. Questions 21 through 25 involve word problems. Each correct multiple choice answer is awarded one point. A score of between 20-25 out of 25 questions was considered above average to very high score by the developers. A score of between 0 and 19 demonstrated a low-to-average level of basic computer related aptitude that researchers have shown to predict success in computer courses.

Scores from this test were used to classify the students as either high or low computer science aptitude within each treatment group. The high aptitude classification was associated with a high potential for success in technical computer courses. These courses often include learning a computer language such as FORTRAN and applying the language in writing computer programs. The low-to-average aptitude classification was used by the university of the developers to permit students to enroll only in non-technical "computer appreciation" courses involving little, if any, computer language and programming training but training in the use of preprogrammed software applications. If students demonstrated success in computer appreciation courses with the use of word processing, spreadsheets, and data bases, then they were allowed to register for increasingly advanced courses.

The reliability and validity of the computer science aptitude test was investigated by Konvalina, Stephens, and Wileman of the computer science program at the University of Nebraska at Omaha. Reliability was performed on the test using the Kuder-Richardson 20 measure of reliability and reported to be 0.76 on page 378 of their research contribution (Konvalina, Wileman, and Stephens, 1983). To measure predictive validity, the test was correlated with a comprehensive final examination and reported to be 0.56 ($p<0.001$; N=165), a highly significant correlation. In addition, the discriminating value of the test between the withdrawal and the nonwithdrawal students in the course under investigation was significant. This was evident in a two-sample t-test between two groups of students (withdrawers=154 and nonwithdrawers=228). The t-test yielded a significance of the difference between the two means. As part of their involvement in computer science education, the developers proposed that the test be used to decide which applicants have a high likelihood of success in computer courses. They also suggested that high-level technical resources can be used more cost-effectively in this way.

The coefficient α reliability of the computer science aptitude test for this study was 0.70. The means were 21.07 and 16.37 for high and low computer science aptitudes, respectively. A t-test yielded a significant difference ($t=9.52$, $df=79$, $n[hi]=27$, $n[lo]=54$).

3.4.2. Treatments

A lesson script was developed by the researcher to control the presentation for each treatment group. The script (see Appendix G, SCRIPT FOR TREATMENTS (HAND-OUT)

AKA LESO'S GUIDE) featured sections 2 through 6 on pages 2 through 14 of the lesson script for the material presented during the treatments. The script is a concise set of instructions or user's guide to accomplish those tasks that represent computer skills necessary to learn computer programming. The three major tasks are: 1. communicate with the computer (that is, logon and logoff the IBM mainframe computer system); 2. prepare programming assignments with the use of the FORTRAN programming language editor WFOR; and 3. secure paper copies of FORTRAN programming assignments from the VM/CMS operating system. The script provides the logical, step-by-step progression of user actions and computer reactions of these tasks. Each of the three major tasks was divided into subtasks that covered commands such as get, add, change, delete, and save program files for the editing task. An instructional systems design model for task analysis (Briggs and Wager, 1981) was used for the preparation of the script. The script was read by the instructor during each treatment. All students received a copy of the script and used it throughout each of the treatments, laboratory exercises, and practical test.

The instructional treatments consisted of three types of presentation strategies (verbal, visual, and hands-on). The verbal presentation strategy was given to section 1 of the Computer Science 201 class. The visual presentation strategy was given to section 2 and the hands-on presentation strategy was given to section 3. All of the treatments were presented by the same instructor on the same day using the same lesson script.

The instructional treatments differed from one another in two ways:

1. The equipment used in the treatments; and
2. The involvement of the students.

The verbal presentation strategy had no computer equipment and the students used passive learning strategies. The instructor read aloud from the script as the students observed their copy of the script.

The visual presentation strategy had a demonstration of the computer skills by the instructor at a computer which projected the output from the computer screen onto a large screen display at the front of the classroom, while the students used passive learning strategies. The instructor read aloud from the script while demonstrating at the computer as the students observed their copy of the script and the large screen display.

The hands-on presentation strategy occurred in the computer laboratory that afforded each student a computer to actively perform the computer skills that were read from the script by the instructor. The instructor did not use a computer or large screen display but was free to walk about the students and monitor the pace of the group as he read aloud the step-by-step procedures from the script while the students performed the computer skills and observed their copy of the script.

The three types of presentation strategies are summarized below.

Verbal Treatment:	Oral presentation of the instructional script by the instructor. Students use words and text from the script and the instructor as mediators.
Visual Treatment:	Computer demonstration by the instructor. Students used additional cues such as visuals projected on the large screen display as well as words and text from the script and the instructor as mediators.
Hands-on Treatment:	Students actively performed computer skills at computers in the laboratory as well as words and text from the script and the instructor as mediators.

3.4.3. Laboratory Exercises

A set of laboratory exercises was developed by the researcher to allow all treatment groups to reach criterion on the computer. The exercises (see Appendix E, EXERCISE BEFORE PRACTICAL TEST) provided all students with use of the computers in the laboratory and rehearsal of computer skills presented in the script. The exercises stressed the practical applications of the computer skills to recognize and correct syntax errors (practical test problem one) and execution errors (practical test problem two). All students satisfactorily completed the exercises and turned in the computer printed outputs of the assignments (exercises) within one 75-minute laboratory period that occurred prior to the practical test.

3.4.4. Criterion Measures

Three types of tests were used as criterion measures in this study. They were the State Anxiety Inventory Form, the Written Achievement Test, and the Practical Achievement Test of Computer Skills.

3.4.4.1. Anxiety Inventory

Computer anxiety was measured using the State Anxiety Inventory Form developed by Spielberger and others (1983). The form, called STAI Form Y-1 (see Appendix C, PRE AND POST TREATMENT QUESTIONNAIRE FOR STATE COMPUTER ANXIETY) was used to measure pre-treatment computer anxiety and post-treatment computer anxiety.

Form Y-1, the state computer anxiety questionnaire, contained 20 statements numbered 1 through 20 which people have used to describe their feelings about computers. For example, "I feel self-confident" when I use a computer. The students were requested to read each statement and indicate how they felt at the moments prior to and following the uses of the computer. One of 4 possible answers could be selected which seemed to best describe the student's present feelings. The answers were: 1. Not at all; 2. Somewhat;

3. Moderately so; and 4. Very much so. The answers were scored on a 4-point scale ranging from 4 points for "Not at all", to 1 point for "Very much so" for statements about positive feelings and the opposite range for statements about negative feelings. The statements were randomly organized between positive and negative feelings to minimize the use of any pattern when selecting the answers. A maximum score of 80 indicated the highest level of computer anxiety. A minimum score of 20 indicated the lowest level of computer anxiety.

A representative measure of reliability for the State Anxiety Inventory Form Y-1 was reported as .92 by Spielberger. The early test construction and validation precedures, carried out primarily with undergraduate college students at Vanderbilt University, were described by Spielberger in the Anxiety Manual, section III., pp. 12-19.

Coefficient α reliability of the State Anxiety Inventory Form Y-1 based on this sample was 0.83.

3.4.4.2. Written Achievement Test

The written achievement test (see Appendix D, WRITTEN ACHIEVEMENT TEST AFTER TREATMENT) was designed by the researcher to evaluate the recall of the computer lesson. The lesson was presented from a lesson script that identified and demonstrated the use of computer skills. The written achievement test was constructed from the set of objectives established within the lesson script. The objectives covered five levels of learning: fact, discrimination, defined concept, rule using, and problem solving. The written test was used to determine what knowledge of computer skills the student remembered from the instructional presentation.

The written test was composed of 6 parts. Each part of the written test was directly related to a specific section or sections of the lesson script. These are mapped below as "THE INFORMATION TESTED IN PART x WAS PRESENTED IN THE SCRIPT SECTION y:

THE INFORMATION TESTED IN	WAS PRESENTED IN SCRIPT
Part 1 (fact)	section 3.
Part 2 (defined concept)	section 2.
Part 3 (defined concept)	section 6.
Part 4 (discrimination)	sections 2-6.
Part 5 (rule)	section 4.
Part 6 (problem solving)	sections 3 and 6.

Part one (fact) of the written test required the student to state the definition of a computer editor. Section 3 of the script covered creating, editing, viewing and saving a file with WFOR.

Part two (defined concept) required the student to list the proper sequence for logging on to the computer, given the five steps. Section 2 of the script covered logging on to the IBM VM/CMS computer system.

Part three (defined concept) required the student to list the proper sequence for running a FORTRAN program and making a listing of output, given the nine steps. Section 6 of the script covered how to make a listing of output generated during a program's run.

Part four (discrimination) required the student to discriminate the types of commands by matching 16 given command names with either CMS or WFOR command type or Not Applicable. Sections 2 through 6 of the script presented the CMS and WFOR commands. Figure 8.0 of the script listed each command by type.

Part five (rule) required the student to classify the type of editing capability as either insert, change, or delete, given descriptions relating to editing a file using WFOR. Section 4 of the script covered editing a saved FORTRAN file.

Part six (problem solving) required the student to write the correct instructions that will generate a FORTRAN program file that executes without error, given a FORTRAN program example with errors. Sections 3 and 6 of the script covered the correct FORTRAN syntax of the instructions in the file and the step-by-step use of the commands to execute the FORTRAN program without error.

A total of 50 points was awarded on the written achievement test. Part one had 2 points; part two had 5; part three had 9; part four had 16; part 5 had 8; and part six had 10.

Coefficient α reliability of the written achievement test was 0.68. Validity of the test was established by the objective of each part and the kind of test behavior (action verb) measured. Both the verbs and the objects (e.g., to classify the type of editing capability) must be correctly used in the test for it to be valid. The validity of the written achievement test was confirmed by senior faculty members of the Instructional Systems Program of Penn State.

3.4.4.3. Practical Achievement Test.

The practical achievement test (see Appendix F, PRACTICAL ACHIEVEMENT TEST ONE WEEK AFTER TREATMENT) was designed by the researcher to evaluate the use of the computer skills at the computer. The practical test represents a more realistic evaluation of performance of computer skills at the computer, whereas the written test evaluated knowledge of what computer skills to use for the desired outcomes.

The practical test consisted of 5 parts which covered the same content areas as did parts 2 through 6 of the written test. Part 1 of the written test was not tested on the practical test. A mapping of the parts of the practical test that coincided with the parts of the written test is presented as PRACTICAL TEST PART x COINCIDED WITH PART y OF THE WRITTEN TEST below:

PRACTICAL TEST	COINCIDED WITH	OF WRITTEN TEST
Part 1		parts 2, 3, 5, and 6.
Part 2		parts 2, 3, 5, and 6.
Part 3		parts 3 and 4.
Part 4		parts 3, 4, and 6.
Part 5		parts 3, 4, and 6.

Part one of the practical test required the student to get a saved FORTRAN program file that contained one syntax error. The syntax error was to be located by viewing the file and changing the line in the file with the error. The correctly edited program file was to be listed off as output to the printer. Part one of the practical test was awarded 10 points and coincided with parts 2, 3, 5, and 6 of the written test.

Part two of the practical test required the student to get a second saved FORTRAN program file that contained one execution error. The execution error was to be debugged by running the program and detecting the problematic instruction. A portion of the problematic instruction was to be deleted and the correction inserted at the proper location. The correctly edited/debugged program file was to be listed off as output to the printer. Part two of the practical test was awarded 10 points and coincided with parts 2, 3, 5, and 6 of the written test.

Part three of the practical test required the student to distinguish between CMS commands and WFOR commands to listoff a directory for the WFOR program files that were edited in parts one and two as well as CMS files contained on the student's disk. Part three of the practical test was awarded 10 points and coincided with parts 3 and 4 of the written test.

Part four of the practical test required the student to get and execute the FORTRAN program file with the corrected syntax error and listoff the correct output to the printer without error. Part four of the practical test was awarded 10 points and coincided with parts 3, 4, and 6 of the written test.

Part five of the practical test required the student to get, debug, and execute the FORTRAN program file with the incorrect output and listoff the correct output to the printer without error. Part five of the practical test was awarded 10 points and coincided with parts 3, 4, and 6 of the written test.

A total of 50 points was awarded on the practical achievement test. Parts one through five were 10 points each.

Coefficient α reliability of the practical test was 0.70. Validity of the practical achievement test was confirmed by faculty members of the Computer Science Department of the Altoona Campus.

3.5. Procedure

The students registered for 3 separate sections of Computer Science 201. Each section was randomly assigned to a type of presentation strategy (verbal, visual, or hands-on). The verbal presentation strategy was given to section 1 of the Computer Science 201 class. The visual presentation strategy was given to section 2 and the hands-on presentation strategy was given to section 3.

All of the presentation strategies were presented by the same instructor on the same day using the same lesson script. Everyone had a copy of the lesson script. The pace of the instruction was controlled by the content of the script. The instructor read the step-by-step procedures to each treatment group, first giving the user action associated with a computer skill, followed by the computer reaction for each task/command to be completed. No questions from the students were allowed during the presentation strategies. Each presentation strategy required 45 minutes of the alotted 75 minutes of class time.

A written achievement test immediately followed the 45 minutes of the presentation strategy to measure specific educational objectives. The test consumed the remaining 30 minutes of class time.

Laboratory computer exercises were assigned at the next class meeting that followed the treatments to allow the students to reach criterion for the practical test. The exercises were completed during a 75 minute laboratory period. At the next class meeting after the exercises a practical achievement test on the computer was administered in the computer laboratory. The test lasted 30 minutes. Immediately after the practical test, the questionnaire on computer anxiety was administered.

For a summary of the time sequence involved with the procedure please refer back to Figure 3.1, Overall Design of Study.

3.6. Data Analysis

The data from the study was analyzed by means of a two factor Multivariate Analysis of Variance (MANOVA). The independent variables were Presentation Strategy and Computer Science Aptitude. Referring to Figure 3.1, Overall Design of Study, Presentation Strategy consisted of 3 levels (verbal, visual, and hands-on), and Computer Science Aptitude consisted of 2 levels (high and low). Two dependent variables were (1) student achievement of computer skills for computer programming as measured by the written test immediately after the treatment and (2) the practical test which was delayed one week. Another dependent variable was post-treatment Computer Anxiety.

Univariate analysis of variance (ANOVA) for each dependent variable can be conducted when the multivariate test statistic is significant. When comparing more than two means, a significant F test indicates that the means are significantly different from each other, but it does not indicate which means differ from which other means. Multiple comparison methods give more detailed information about the difference among the means.

One approach to multiple comparisons is the pairwise comparison of means. Pairwise comparisons of the means for all main effects was performed by the Bonferroni method.

For all statistical analyses and multiple comparisons tests, the Type I error probability was set at 0.05.

Chapter 4

STATISTICAL ANALYSIS AND RESULTS

Data were collected from 81 freshman and sophomore college engineering students from the Altoona Campus of The Pennsylvania State University. The students registered for 3 sections of an introductory programming course for engineers titled Computer Science 201. Each of the sections had an enrollment of 27 students. Section 1 was the VERBAL PRESENTATION STRATEGY group, section 2 was the VISUAL PRESENTATION STRATEGY group, and section 3 was the HANDS-ON PRESENTATION STRATEGY group.

During the third week of the fifteen-week semester, which was two classes prior to the training with different presentation strategies (treatments), all students received the computer science aptitude test. Results from this test separated the students into 2 groups representing those students who scored a HIGH APTITUDE potential for learning computer skills and those students who scored a LOW APTITUDE potential for learning computer skills.

One class prior to the treatments all students completed the State Anxiety Inventory self-evaluation questionnaire. The event recorded "how they felt, at that moment," just before their first use of the computer in the course. The questionnaire measured PRE-TREATMENT COMPUTER ANXIETY.

The next class meeting for the 3 groups consisted of the training of the 3 different presentation strategies (VERBAL, VISUAL, and HANDS-ON) delivered by the same instructor. Each of the 3 groups received 45 minutes of instruction followed by a WRITTEN ACHIEVEMENT TEST. One week later, all students completed a PRACTICAL ACHIEVEMENT TEST at the computer. The written achievement test measured "what" computer skills to use while the practical test measured "performance" of computer skills. After the practical test and while still at the computer all students completed the POST-TREATMENT COMPUTER ANXIETY questionnaire.

4.1. Descriptive Statistics

Table 4.1 summarizes the three presentation strategy groups (verbal, visual and hands-on) by high and low computer science aptitude groups. Reported for each group is actual frequency and percent. The actual frequency represents the actual number of students (n) per group. Totals for each row and column are also reported.

Table 4.1 Table of Presentation Strategy by Aptitude

Class Level Information

Class	Levels	Values		
PRESENT. STRATEGY	3	Verbal	Visual	Hands-on
APTITUDE	2	Low	High	

Number of observations in data set = 81.

PRESENT STRATEGY

Cell Size (n)	APTITUDE Low	APTITUDE High	Total
Verbal	15	12	27
Visual	21	6	27
Hands-on	18	9	27
Total	54	27	81

For the verbal presentation strategy treatment, low computer science aptitude group, n = 15. For the verbal presentation strategy treatment, high computer science aptitude group, n = 12.

For the visual presentation strategy treatment, low computer science aptitude group, n = 21. For the visual presentation strategy treatment, high computer science aptitude group, n = 6.

For the hands-on presentation strategy treatment, low computer science aptitude group, n = 18. For the hands-on presentation strategy treatment, high computer science aptitude group, n = 9.

Each of the three presentation strategy treatment levels had a balanced total of 27 students for a sample size of 81 students in the experiment.

Tables 4.2 through 4.5 present the descriptive statistics for the groups in the study. They summarize the scores for each level of presentation strategy (hands-on, visual, and verbal), for each level of computer science aptitude (high and low), and for each of the 6 groups of the two-way, 3-by-2 factorial model. Reported is the group size (n), the mean score (mean), the standard deviation (std dev), the minimun score (min), and the maximum score (max). These results are reported for the three dependent variables consisting of the written achievement test (Table 4.2), the practical test (Table 4.3), and post-treatment computer anxiety (Table 4.4). Note that the descriptive statistics for pre-treatment computer anxiety were included in Table 4.5. There were no significant differences in mean pre-treatment computer anxiety scores among the three presentation strategies. A separate ANOVA test reported an F value of 0.11 and Pr > F of 0.89. An interesting pattern to observe among the groups was the higher post-treatment computer anxiety means (Table 4.4) from the pre-treatment computer anxiety means (Table 4.5). Post-treatment computer anxiety means were 5 points higher than pre-treatment computer anxiety means.

4.2 General Analysis Methods

The pre-treatment and post-treatment computer anxiety questionnaires, the computer science aptitude test, the written achievement test, and the practical test were hand scored by the investigator. All data were collected on spreadsheet and converted to a computer file by the investigator. Statistical analyses were performed in SAS (Statistical Analysis System).

The General Linear Models (GLM) Procedure of SAS was used to execute the multivariate analysis of variance. The independent variables (or "class" variables as reported by SAS) were PRESENTATION STRATEGY and COMPUTER SCIENCE APTITUDE. The PRESENTATION STRATEGY class had 3 levels called VERBAL, VISUAL, and HANDS-ON PRESENTATION STRATEGY. The COMPUTER SCIENCE APTITUDE class had 2 levels called HIGH and LOW COMPUTER SCIENCE APTITUDE.

Table 4.2 Descriptive Statistics for Written Achievement Test

Group		N	Mean	Std Dev	Min	Max
Total		81	34.38	6.89	17	48
Level of Strategy	Level of Aptitude					
Hands-on		27	35.18	7.73	18	47
Visual		27	35.25	7.00	26	48
Verbal		27	32.70	5.75	17	45
	High	27	35.59	7.05	23	48
	Low	54	33.77	6.80	17	45
Hands-on	High	9	36.77	7.74	23	47
Hands-on	Low	18	34.38	7.83	18	45
Visual	High	6	38.83	9.36	29	48
Visual	Low	21	34.23	6.07	26	46
Verbal	High	12	33.08	4.52	25	39
Verbal	Low	15	32.40	6.71	17	45

Table 4.3 Descriptive Statistics for Practical Test

Group		N	Mean	Std Dev	Min	Max
Total		81	30.24	17.17	0	50
Level of Strategy	Level of Aptitude					
Hands-on		27	39.25	11.74	10	50
Visual		27	21.48	16.33	0	50
Verbal		27	30.00	18.39	0	50
	High	27	30.74	17.95	0	50
	Low	54	30.00	16.93	0	50
Hands-on	High	9	37.80	13.00	10	50
Hands-on	Low	18	40.00	11.40	10	50
Visual	High	6	18.30	19.40	0	50
Visual	Low	21	22.40	15.80	0	50
Verbal	High	12	31.70	18.50	0	50
Verbal	Low	15	28.70	18.84	0	50

Table 4.4 Descriptive Statistics for Post-Treatment Computer Anxiety

Group		N	Mean	Std Dev	Min	Max
Total		81	45.40	12.73	22	80
Level of Strategy	Level of Aptitude					
Hands-on		27	45.07	10.01	25	75
Visual		27	45.74	15.19	23	79
Verbal		27	45.40	12.94	22	80
	High	27	42.44	11.82	22	80
	Low	54	46.88	13.01	23	79
Hands-on	High	9	40.55	4.79	33	49
Hands-on	Low	18	47.33	11.20	25	75
Visual	High	6	44.16	13.52	27	61
Visual	Low	21	46.19	15.92	23	79
Verbal	High	12	43.00	14.99	22	80
Verbal	Low	15	47.33	11.20	27	64

Table 4.5 Descriptive Statistics for Pre-Treatment Computer Anxiety

Group		N	Mean	Std Dev	Min	Max
Total		81	40.19	12.01	20	70
Level of Strategy	Level of Aptitude					
Hands-on		27	40.55	10.34	22	60
Visual		27	40.29	15.00	20	70
Verbal		27	39.74	10.57	23	67
	High	27	37.40	9.58	22	59
	Low	54	41.59	12.92	20	70
Hands-on	High	9	37.33	10.45	22	59
Hands-on	Low	18	42.17	10.19	28	60
Visual	High	6	35.67	10.89	23	52
Visual	Low	21	41.62	15.97	20	70
Verbal	High	12	38.33	9.00	23	55
Verbal	Low	15	40.87	11.87	25	67

The MANOVA consisted of a two-way, factorial model with interaction. The three dependent variables were the WRITTEN ACHIEVEMENT TEST, the PRACTICAL TEST, and POST-TREATMENT COMPUTER ANXIETY. ANOVA tables for each dependent variable were also computed. Results of univariate analysis of variance are appropriate when the multivariate analysis is significant.

Multiple comparisons among the three presentation strategies were conducted using the Bonferroni method. For all statistical analyses and multiple comparisons tests, the Type I error probability was set at 0.05.

Table 4.6 presents the results of the MANOVA for the main effects of PRESENTATION STRATEGY (VERBAL, VISUAL, and HANDS-ON) and COMPUTER SCIENCE APTITUDE (HIGH and LOW). The interaction effect of PRESENTATION STRATEGY by COMPUTER SCIENCE APTITUDE is also presented.

The multivariate test statistics reported Wilks' Lambda. The results of the MANOVA for the effects of presentation strategy (verbal, visual, and hands-on) and computer science aptitude (high and low) indicated "no Overall PRESENTATION STRATEGY by APTITUDE Effect" and "no Overall APTITUDE Effect" on the three dependent variables of written achievement test, practical test, and post-treatment computer anxiety. Research hypotheses H3 (Computer science aptitude on written test) and H4 (Computer science aptitude on post-treatment computer anxiety) were not supported (see sections 4.5 and 4.6, respectively). Neither was the interaction effect of research hypothesis H5i (Presentation strategy with computer science aptitude on practical test) supported (see section 4.7).

The results of the MANOVA did show a significant main effect for presentation strategy as reported by the F statistic for Wilks' Lambda ($F = 3.475$; $Pr > F = 0.0031$). Research hypotheses H1 (Presentation strategy on written test), H2 (Presentation strategy on post-treatment computer anxiety), and H5m-main effect (Presentation strategy on practical test) were investigated further (continued on next page plus sections 4.4 and 4.7, respectively).

4.3. Analysis of Presentation Strategy on Written Test

The null hypothesis and the two research hypotheses for presentation strategy on written test were:

H1.0 There will be no significant differences in mean written achievement test scores among three presentation strategies.

H1.1 Mean written achievement test scores for hands-on training are higher than mean written achievement test scores for visual training.

H1.2 Mean written achievement test scores for visual training are higher than mean written achievement test scores for verbal training.

Table 4.6 (MANOVA) Multivariate Analysis of Variance

Manova Test Criteria and F Approximations for
the Hypothesis of no Overall PRESENTATION STRATEGY Effect
H = Type III SS&CP Matrix for PRESENT. STRATEGY E = Error SS&CP Matrix

S=2 M=0 N=35.5

Statistic	Value	F	Num DF	Den DF	Pr > F
Wilks' Lambda	0.765650967	3.47572	6	146	**** 0.0031

Manova Test Criteria and Exact F Statistics for
the Hypothesis of no Overall APTITUDE Effect
H = Type III SS&CP Matrix for APTITUDE E = Error SS&CP Matrix

S=1 M=0.5 N=35.5

Statistic	Value	F	Num DF	Den DF	Pr > F
Wilks' Lambda	0.938512724	1.59421	3	73	0.1981

Manova Test Criteria and F Approximations for
the Hypothesis of no Overall PRESENTATION STRATEGY*APTITUDE Effect
H = Type III SS&CP Matrix for STRATEGY*APTITUDE E = Error SS&CP Matrix

S=2 M=0 N=35.5

Statistic	Value	F	Num DF	Den DF	Pr > F
Wilks' Lambda	0.968176905	0.39667	6	146	0.8802

NOTE: F Statistic for Wilks' Lambda is exact.

$p < 0.05$ ****

A SAS Type III sum of squares was used for this study because of unequal cell frequencies. Table 4.7 presents the ANOVA tests of significance for the written achievement test. The results of the univariate analysis for the written achievement test indicated no significant main effects for either presentation strategy or computer science aptitude. No significant two-way interaction effect existed as well for the written achievement test. Research hypotheses H1.1 and H1.2 stated in terms of the main effect for presentation strategy on written test were not supported. No statistical differences existed among the mean written achievement test scores for hands-on, visual, and verbal training.

4.4 Analysis of Presentation Strategy on Post-Treatment Computer Anxiety

The null hypothesis and the two research hypotheses for presentation strategy on post-treatment computer anxiety were:

H2.0 There will be no significant differences in mean post-treatment computer anxiety scores among three presentation strategies.

H2.1 Mean post-treatment computer anxiety scores for the hands-on presentation strategy are lower than mean post-treatment computer anxiety scores for visual and verbal presentation strategies.

Table 4.7 Tests of Significance for Written Test

Univariate Analysis of Variance (ANOVA)

Dependent Variable: WRITTEN ACHIEVEMENT TEST

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	250.14295	50.02859	1.06	0.3919
Error	75	3554.99286	47.39990		
Corrected Total	80	3805.13580			

R-Square	C.V.	Root MSE	WRITTEN TEST Mean
0.065738	20.02389	6.8848	34.383

Source	DF	Type III SS	Mean Square	F Value	Pr > F
PRESENTATION STRATEGY	2	183.74717	91.87359	1.94	0.1511
APTITUDE	1	110.72546	110.72546	2.34	0.1306
PRES. STRATEGY*APTITUDE	2	42.04347	21.02173	0.44	0.6435

H2.2 Visual and verbal presentation strategies have a common post-treatment computer anxiety mean.

Table 4.8 presents the tests of significance for post-treatment computer anxiety. The results of the ANOVA for post-treatment computer anxiety indicated no significant main effects for either presentation strategy or computer science aptitude. Research hypothesis H2.1 was not supported. Mean post-treatment computer anxiety scores were not significantly different between the hands-on and visual or verbal presentation strategies.

Research hypothesis H2.2 was supported. The null hypothesis H2.0 failed to be rejected. The visual and verbal presentation strategy groups did have a common post-treatment computer anxiety mean score as expected (see Table 4.4 for reported means of 45.74 and 45.40, respectively). The mean post-treatment computer anxiety scores for visual (45.74) and verbal (45.40) presentation strategies are common as expected from research hypothesis H2.2. The mean post-treatment computer anxiety score of 45 also for the hands-on presentation strategy group was unexpected. No significant two-way interaction effect existed for post-treatment computer anxiety.

Table 4.8 Tests of Significance for Computer Anxiety

Univariate Analysis of Variance (ANOVA)

Dependent Variable: POST-TREATMENT COMPUTER ANXIETY

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	425.92857	85.18571	0.51	0.7684
Error	75	12545.62698	167.27503		
Corrected Total	80	12971.55556			

R-Square	C.V.	Root MSE	ANXIETY Mean
0.032836	28.48320	12.933	45.407

Source	DF	Type III SS	Mean Square	F Value	Pr > F
PRESENTATION STRATEGY	2	23.63156	11.81578	0.07	0.9319
APTITUDE	1	324.93712	324.93712	1.94	0.1675
PRES. STRATEGY*APTITUDE	2	59.87088	29.93544	0.18	0.8365

4.5 Analysis of Computer Science Aptitude on Written Test

The null hypothesis and the research hypothesis for computer science aptitude on written test were:

H3.0 There will be no significant differences in mean written test scores of students with two levels of computer science aptitude.

H3.1 The mean written test score of students with high computer science aptitude is higher than the mean written test score of students with low computer science aptitude.

Table 4.7 presented the ANOVA tests of significance for the written achievement test. The results of the univariate analysis for the written achievement test indicated no significant main effects for either computer science aptitude or presentation strategy. No significant two-way interaction effect existed as well for the written achievement test. Research hypothesis H3.1 was not supported. No statistical difference existed between the mean written test scores of students with high and low computer science aptitudes. The multivariate analysis of variance presented in Table 4.6 reported no overall aptitude effect.

4.6 Analysis of Computer Science Aptitude on Post-Treatment Computer Anxiety

The null hypothesis and the research hypothesis for computer science aptitude on post-treatment computer anxiety were:

H4.0 There will be no significant differences in mean post-treatment computer anxiety scores of students with two levels of computer science aptitude.

H4.1 The mean post-treatment computer anxiety score of students with high computer science aptitude is lower than the mean post-treatment computer anxiety score of students with low computer science aptitude.

Table 4.8 presented the tests of significance for post-treatment computer anxiety. The results of the ANOVA for post-treatment computer anxiety indicated no significant main effects for either computer science aptitude or presentation strategy. Research hypothesis H4.1 was not supported. Mean post-treatment computer anxiety scores were not significantly different between students with high and low computer science aptitudes. The multivariate analysis of variance presented in Table 4.6 reported no overall aptitude effect.

4.7 Analysis of Presentation Strategy and Computer Science Aptitude on Practical Test

The null hypothesis and the research hypotheses for the interaction effect for presentation strategy by computer science aptitude on practical test were:

H5i.0 There will be no two-way interaction between presentation strategy and computer science aptitude on the practical test.

H5i.1 College students with high and low computer science aptitude and trained by hands-on presentation strategy have a higher mean practical test score than any other combination of the two factors.

H5i.2 The mean practical test scores for hands-on-high and hands-on-low groups are equal.

H5i.3 The mean practical test scores for the visual-high and verbal-high groups are higher than the mean practical test scores for the visual-low and verbal-low groups.

The null hypothesis and the research hypotheses for the main effect for presentation strategy on practical test were:

H5m.0 There will be no significant differences in mean practical test scores among three presentation strategies.

H5m.4 Mean practical test scores for the hands-on presentation strategy are higher than mean practical test scores for the visual presentation strategy.

H5m.5 Mean practical test scores for the hands-on presentation strategy are higher than mean practical test scores for the verbal presentation strategy.

Table 4.9 presents the tests of significance for the practical achievement test. The results of the ANOVA for the practical achievement test did show a significant main effect for presentation strategy. No significant main effect for computer science aptitude and no significant two-way interaction effect were evident for the practical achievement test. Research hypotheses H5i.1 and H5i.3 stated in terms of the two-way interaction effect were not supported for the practical test. Students with high and low computer science aptitudes and trained by hands-on presentation strategy did not have significantly higher mean practical test scores than any other combination of the two factors. Students with high computer science aptitude and trained by visual presentation strategy had the smallest mean practical test score among the verbal-high, visual-low, and verbal-low groups. The multivariate analysis of variance presented in Table 4.6 reported no overall presentation strategy by aptitude effect.

Research hypothesis H5i.2 was supported. The null hypothesis H5i.0 failed to be rejected. The hands-on-high group and the hands-on-low group did have a common practical test score mean as expected. Table 4.3 reported practical test score means of 37.8 and 40 for hands-on-high and hands-on-low groups, respectively.

Table 4.9 Tests of Significance for Practical Test

Univariate Analysis of Variance (ANOVA)

Dependent Variable: PRACTICAL ACHIEVEMENT TEST

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	4435.2205	887.0441	3.47	**** 0.0071
Error	75	19159.8413	255.4646		
Corrected Total	80	23595.0617			

	R-Square	C.V.	Root MSE	PRACT. TEST Mean
	0.187972	52.84261	15.983	30.247

Source	DF	Type III SS	Mean Square	F Value	Pr > F
PRESENTATION STRATEGY	2	3612.1673	1806.0836	7.07	**** 0.0015
APTITUDE	1	20.1371	20.1371	0.08	0.7797
PRES. STRATEGY*APTITUDE	2	157.4664	78.7332	0.31	0.7357

$p < 0.05$ ****

No interaction effect for research hypothesis H5i.3 was supported by the univariate analysis for the practical test. Students with high computer science aptitude and trained by visual and verbal presentation strategies did not have significantly higher mean practical test scores than students with low computer science aptitude and trained by visual and verbal presentation strategies.

A main effect for presentation strategy on the practical test was indicated by the results of the univariate analysis. The F statistic for presentation strategy was reported as 7.07 and the probability greater than F was 0.0015. Thus the null hypothesis H5m.0 was rejected and support was found for the main effect for presentation strategy on practical test. The ANOVA F-test indicated that the means of the practical test for presentation strategies are significantly different from each other. Hence a multiple comparison T-test was necessary to provide more detailed information about the differences among the means for the 3 presentation strategies (verbal, visual, and hands-on).

Table 4.10 presents the univariate T-tests of differences between means for the 3 levels of presentation strategies (verbal, visual, and hands-on). A multiple comparisons t-test performed by the Bonferroni (Dunn) T-test procedure for the dependent variable practical achievement test indicated a significant difference between the hands-on presentation strategy and the visual presentation strategy

Table 4.10 Univariate T-Tests of Differences Between Means

Presentation Strategies (Verbal, Visual, and Hands-on)
Bonferroni (Dunn) T-tests for variable: PRACTICAL TEST

Alpha= 0.05 Confidence= 0.95 df= 75 MSE= 255.4646
Critical Value of T= 2.44886
Minimum Significant Difference= 10.653

Comparisons significant at the 0.05 level are indicated by '***'

PRESENTATION STRATEGY Comparison	Simultaneous Lower Confidence Limit	Difference Between Means	Simultaneous Upper Confidence Limit	
hands-on - verbal	-1.394	9.259	19.912	
hands-on - visual	7.125	17.778	28.431	***
verbal - hands-on	-19.912	-9.259	1.394	
verbal - visual	-2.134	8.519	19.171	
visual - hands-on	-28.431	-17.778	-7.125	***
visual - verbal	-19.171	-8.519	2.134	

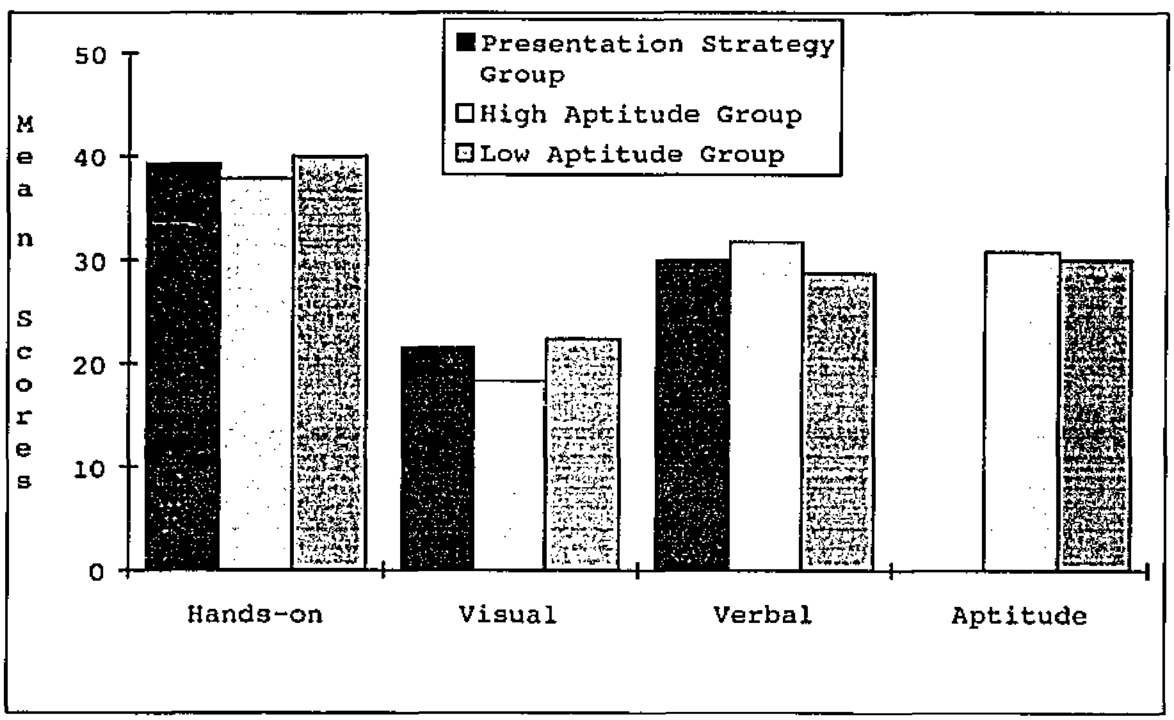


Figure 4.1 Practical Test Mean Scores

(difference between means = 17.778; minimum significant difference = 10.653; critical value of T = 2.448; df = 75; alpha = 0.05). The result presented in Figure 4.1 leads to the inference that the hands-on presentation strategy yields a higher mean practical test score than the visual presentation strategy. The main effect of research hypothesis H5m.4 was supported for means for hands-on training being higher than visual training, however support was not provided for the main effect of research hypothesis H5m.5 for means for hands-on training being higher than verbal training. The difference between means for hands-on presentation strategy and verbal presentation strategy was reported as 9.259, which was very close to the minimum significant difference of 10.653, but it was not significant. The difference between means for verbal presentation strategy and visual presentation strategy was reported as 8.519.

A summary of the mean scores and standard deviations by groups composed of the 3 main levels of presentation strategies (verbal, visual, and hands-on) and 2 main levels of computer science aptitude (low and high) was presented in Tables 4.2 - 4.4. The significant main effect for presentation strategy on practical test from research hypothesis H5m.4 is confirmed by the mean score of 39.26 for the hands-on group which exceeds the mean score of 21.48 for the visual group by 8 points over the minimum significant difference of 10.65. The difference between the hands-on group and the verbal group was lacking 1.4 points of matching the minimum significant difference. Thus the mean score of 39.26 for the hands-on group did not significantly exceed the minimum difference of 10.65 when compared with the mean score of 30.0 for the verbal group. Hence there was no support for research hypothesis H5m.5 main effect. The significant difference only applies to the hands-on presentation strategy having a higher mean practical test score from the visual presentation strategy.

Chapter 5

FINDINGS, CONCLUSIONS, and RECOMMENDATIONS

5.1. Objective, Design, and Hypotheses

The objective of this research was to investigate the relative effectiveness of three presentation strategies (verbal, visual, and hands-on) and their effects on student achievement of computer skills necessary for computer programming and on computer anxiety. Another intention of this study was to ascertain whether or not computer science aptitude had an effect on student achievement of computer skills and on computer anxiety and interacted with presentation strategies.

The overall design of this study was presented in Figure 3.1. Data were collected from 81 freshman and sophomore college engineering students who registered for three separate sections of Computer Science 201 during their spring semester at Penn State Altoona Campus. All of the students received the same lesson script in prose format but the presentation strategies were different among the three sections. The 27 students enrolled in CmpSci 201, section 1, were trained with a verbal presentation strategy. The 27 students enrolled in CmpSci 201, section 2, were trained with a visual presentation strategy. And the 27 students enrolled in CmpSci 201, section 3, were trained with a hands-on presentation strategy.

All of the students were classified as either high aptitude or low aptitude for introductory, technical computer science course work based on a computer science aptitude test that was administered at the outset of the study. A computer anxiety questionnaire was completed by all of the students one class prior to the presentation strategy treatments and accounted for the pre-test computer anxiety levels. A written achievement test was administered immediately after each treatment of a verbal, visual, and hands-on presentation strategy to evaluate recall of "what" computer skills to use when programming a computer. A practical achievement test conducted on the computer was delayed one week so that all students could perform computer laboratory exercises in order to reach criterion with practical use of computer skills on computers. The practical achievement test was administered in the computer laboratory to evaluate performance of computer skills while at the computer that are necessary for programming computers. Directly after the completion of the practical test and while still seated at the computers in the computer laboratory, all of the students completed a post-test computer anxiety questionnaire that measured how they felt at that moment immediately after computer use.

The original research hypotheses and a summary of the statistical results based on analyses of variance and appropriate multiple comparison t-tests are provided in the following Tables 5.1 and 5.2.

Table 5.1 Review of Table of Presentation Strategy by Aptitude

Class Level Information

Class	Levels	Values		
PRESENT. STRATEGY	3	Verbal	Visual	Hands-on
APTITUDE	2	Low	High	

Number of observations in data set = 81.

APTITUDE

PRESENT STRATEGY: Cell Size (n)	Low	High	Total
Verbal	15	12	27
Visual	21	6	27
Hands-on	18	9	27
Total	54	27	81

Table 5.2. Statistical Results Relating to Hypotheses

H1. Presentation strategy on written test:
H1.0==> Null Hypothesis (Hands-on=Visual=Verbal) is retained.
H1.1==> Research Hypothesis (Hands-on > Visual) is rejected.
H1.2==> Research Hypothesis (Visual > Verbal) is rejected.

H2. Presentation strategy on post-treatment computer anxiety:
H2.0==> Null Hypothesis (Hands-on=Visual=Verbal) is retained.
H2.1==> Research Hypothesis (Hands-on < Visual) is rejected.
H2.2==> Research Hypothesis (Visual = Verbal) is retained.

H3. Computer science aptitude on written test:
H3.0==> Null Hypothesis (High = Low) is retained.
H3.1==> Research Hypothesis (High > Low) is rejected.

H4. Computer science aptitude on post-treatment computer anxiety:
H4.0==> Null Hypothesis (High = Low) is retained.
H4.1==> Research Hypothesis (High < Low) is rejected.

H5i Presentation strategy*computer science aptitude on practical test:
H5i.0==> Null Hypothesis (HoH=HoL=ViH=ViL=VeH=VeL) is retained.
H5i.1==> Research Hypothesis (HoH, HoL > ViH, ViL, VeH, VeL) is rejected.
H5i.2==> Research Hypothesis (HoH = HoL) is retained.
H5i.3==> Research Hypothesis (ViH, VeH > ViL, VeL) is rejected.

H5m Presentation strategy on practical test:
H5m.0==> Null Hypothesis (Hands-on=Visual=Verbal) is rejected.
H5m.4==> Research Hypothesis (Hands-on > Visual) is retained.
H5m.5==> Research Hypothesis (Hands-on > Verbal) is rejected.

Table 5.1 provides a review of the 2 factors, presentation strategy and computer science aptitude, and the 3-by-2 factorial design. Table 5.2 summarizes the results presented in chapter 4 with respect to the original hypotheses stated in chapter 1.

5.2. Interpretation of the Results

The results of MANOVA for the effects of presentation strategy (verbal, visual, and hands-on) and computer science aptitude (high and low) on the dependent variables of written achievement test, practical achievement test, and post-treatment computer anxiety were presented in Table 4.8. A significant main effect existed for presentation strategy on the practical achievement test as reported in Table 4.10. The other effects under investigation, namely, computer science aptitude and a two-way interaction effect for presentation strategy and computer science aptitude, were not significant. Interpretation of these results will be presented in the following sections based on the hypotheses H1 through H5 presented in section 1.5.

5.2.1. Presentation Strategy on Written Test

H1.0 There will be no significant differences in mean written achievement test scores among three presentation strategies.

H1.1 Mean written achievement test scores for hands-on training are higher than mean written achievement test scores for visual training.

H1.2 Mean written achievement test scores for visual training are higher than mean written achievement test scores for verbal training.

The null hypothesis failed to be rejected. The research hypotheses were rejected for all types of presentation strategies on the written achievement test. The average achievement score for the hands-on presentation strategy group was the same as the average score for the visual presentation strategy group. The average achievement score for visual presentation strategy group was slightly greater than the average score for verbal presentation strategy group, but it was not a statistically significant difference.

The average scores on the written achievement test for hands-on, visual, and verbal presentation strategies were not significantly different. Research of Dwyer (1985), Dale (1985), and Finn (1953) concluded that an increase in the number of cues in the learning environment improves the learning outcomes. More realistic cues represented by visualization and hands-on experience were expected to increase achievement scores over the "control" verbal training strategy. However, the increases were not significant, possibly due to factors held constant among the 3 presentation strategies. That is, the lesson design incorporated task analysis with elaboration strategies, concise step-by-step instructions in prose format, and externally-paced instruction led by the same enthusiastic instructor. Another possible explanation for the small differences could be the result of the type of evaluation. The written achievement test evaluated domain specific knowledge of "what" computer skills to use so actual performance of computer skills was not being evaluated. The results from the evaluation of a written test demonstrated only small differences in subject-matter knowledge, however no pretest was given to determine initial knowledge levels. The written test did not evaluate the performance of the computer skills and problem solving which was more congruent with the objectives of the lesson and the emphasis of the different presentation strategies.

5.2.2. Presentation Strategy on Post-Treatment Computer Anxiety

H2.0 There will be no significant differences in mean post-treatment computer anxiety scores among three presentation strategies.

H2.1 Mean post-treatment computer anxiety scores for the hands-on presentation strategy are lower than mean post-treatment computer anxiety scores for visual and verbal presentation strategies.

H2.2 Visual and verbal presentation strategies have a common post-treatment computer anxiety mean.

The null hypothesis failed to be rejected. Research hypothesis H2.1 was rejected for all types of presentation strategies on post-treatment computer anxiety. The average post-treatment computer anxiety scores for verbal, visual, and hands-on presentation strategy groups were the same. The research hypothesis H2.2 was retained.

The expectation that certain training methods, like those that employ the use of hands-on experience (Howard, Murphy, and Thomas; 1987), or those that provide for successful computer contact (Honeymen and White; 1987), or those that provide for new computer experience (Jones and Wall; 1985), to affect computer anxiety was not supported. The anxiety levels were common among the verbal, visual, and hands-on training methods following the 1-week experiment. Perhaps an elapsed time that was longer than a week for the treatments of different presentation strategies in this study could have had more impact on performance and stress. Fitts and Posner (1976) recommend that skills training include 3 phases of learning: cognitive, associative, and autonomous phases, where performance and stress are considered a function of days or months of practice under a particular training method.

5.2.3. Computer Science Aptitude on Written test

H3.0 There will be no significant differences in mean written test scores of students with two levels of computer science aptitude.

H3.1 The mean written test score of students with high computer science aptitude is higher than the mean written test score of students with low computer science aptitude.

The null hypothesis failed to be rejected. The research hypothesis was rejected for the two levels of computer science aptitude on the written achievement test. The average achievement score on the written test for the high level computer science aptitude group was only slightly greater (1.9 points) than the average score for the low level computer science aptitude group. One explanation for this small difference may be that the handout, or lesson script, which all students received, provided the same written format of the test questions involving concepts, rules, and discriminations on the written achievement test. Whereas the computer science aptitude test, which did have significant mean differences between the high and low groups, involved problem solving only.

Another explanation may be associated with the small dispersion of computer science aptitude raw test scores about the prescribed division point (19) separating the high computer science aptitude group (with raw scores of 20 through 25) and the low computer science aptitude group (with raw scores of 0 through 19). A mean computer science aptitude raw test score of 17.9 was very close to the instrument's division point (19) used in this study. Also, 79 percent of the students scored between 15 and 21 on the computer science aptitude test which produced a very small clustering of scores around the mean (30% below and 49% above). Had the mean score of this study (17.9) been used as the division point instead of the instrument's division point (19), perhaps a more equal distribution of subjects per cell for this sample would have increased the accuracy of the statistical analysis and possibly provided for larger differences when measuring achievement on written tests.

5.2.4. Computer Science Aptitude on Post-Treatment Computer Anxiety

H4.0 There will be no significant differences in mean post-treatment computer anxiety scores of students with two levels of computer science aptitude.

H4.1 The mean post-treatment computer anxiety score of students with high computer science aptitude is lower than the mean post-treatment computer anxiety score of students with low computer science aptitude.

The null hypothesis failed to be rejected. The research hypothesis was rejected for the two levels of computer science aptitude on post-treatment computer anxiety. The average post-treatment computer anxiety score for the high computer science aptitude group (42.4) was less than the average score for the low computer science aptitude group (46.9) as expected, but not significantly so.

Konvalina et al. (1983) and Dwyer (1978) reported that aptitude is one of the most important individual difference variables for the study of learning outcomes. Marcoulides (1988) and McCormick and Ross (1990) substantiated the importance of aptitude when they found that computer anxiety was influenced by past computer experiences. In this study, however, aptitude was not found to be a significant factor.

There is evidence that indicated negative correlations (-0.2) existed between computer science aptitude raw test scores and post-treatment computer anxiety scores. See section 5.4 and Table 5.3 (Correlation) for the results. A high positive correlation (0.72) existed between post-treatment computer anxiety and pre-treatment computer anxiety. The high positive correlation between the pre- and the post-treatment computer anxiety scores indicated the pre-treatment computer anxiety variable may be an appropriate covariate in the analysis of covariance on the dependent variable post-treatment computer anxiety.

5.2.5. Presentation Strategy And Computer Science Aptitude on Practical Test

H5i.0 There will be no two-way interaction between presentation strategy and computer science aptitude on the practical test.

H5i.1 College students with high and low computer science aptitude and trained by hands-on presentation strategy have a higher mean practical test score than any other combination of the two factors.

H5i.2 The mean practical test scores for hands-on-high and hands-on-low groups are equal.

H5i.3 The mean practical test scores for the visual-high and verbal-high groups are higher than the mean practical test scores for the visual-low and verbal-low groups.

H5m.0 There will be no significant differences in mean practical test scores among three presentation strategies.

H5m.4 Mean practical test scores for the hands-on presentation strategy are higher than the mean practical test scores for the visual presentation strategy.

H5m.5 Mean practical test scores for the hands-on presentation strategy are higher than the mean practical test scores for the verbal presentation strategy.

The null hypothesis H5i.0 failed to be rejected for a two-way interaction effect between presentation strategy and computer science aptitude on the practical test. However, a significant main effect for presentation strategy existed on the practical test. The null hypothesis H5m.0 was rejected. Followup t-tests for presentation strategy pairwise comparisons indicated a significant difference between the hands-on presentation strategy and the visual presentation strategy on the practical test. The average score on the practical test for the hands-on presentation strategy group was significantly greater than the average score for the visual presentation strategy group. This result supported research hypothesis H5m.4. The average score for the hands-on presentation strategy group was greater than the average score for the verbal presentation strategy group, although not statistically significant. Hence, research hypothesis H5m.5 was not supported.

The fact that the hands-on presentation strategy group achieved a higher average score for the performance of computer skills as measured by the practical test supported research hypothesis H5m.4 under investigation. The hands-on presentation strategy provided for a significantly more effective training method in acquiring and retaining the use of computer skills on the computer than did the visual presentation strategy. Other researchers lend support to this finding. Banks and Havice (1989) reported that the more realistic the presentation strategy is, the better the opportunity for mastering computer skills. They also suggested that in presenting computer skills, lessons should follow a step-by-step approach. Weinstein and Mayer (1983) indicated that learning is enhanced when the learner possesses a great deal of domain-specific knowledge. Simon (1980) suggested that attention should be given to both subject-matter knowledge and general skills. Wittrock and Lumsdaine (1977) reported that learning is more effective when the training method and learner ability are congruent. Cronbach (1974) emphasized that a training strategy should make available the cues that are to be used in applying the skill when the teaching of a skill is undertaken. Fitts and Posner (1967) suggested realistic experiences be provided in the learning of skills. Dwyer (1985) reported active rehearsal strategy to be more effective than passive. Support from the realism continuum (Dale, 1954; and Finn, 1953) is also evident in the finding of this study.

The hands-on presentation strategy was not more effective than the verbal presentation strategy on the practical test. Only an insignificant difference existed between the average score for the hands-on presentation strategy group and the average score for the verbal presentation strategy group. The association of skills and improved performance of skills vary with time and complexity of tasks. If a longer period of time was provided for the treatments of different presentation strategies, perhaps varying differences would be the results among the average scores on the practical test.

The visual presentation strategy group recorded a lower average achievement score on the practical test than the verbal presentation strategy group. Possibly less demand was place on the college students (Fitts and Posner, 1967) with the visual, demonstration strategy, combined with low-level lighting in the classroom, so that some of the students may have lost alertness, or even gone to sleep during the visual presentation method. Another possibility is that the visual presentation strategy group may have been adversely affected by multiple channel communication interference (Hsia, 1971; Dwyer, 1978) when several sources of stimuli competed with one another. Also, the visual presentation strategy, unlike the other two strategies, did not permit instructor-to-student eye contact during the training session, nor did it allow for the instructor to move about the room freely since he was confined to the computer terminal.

The average achievement score on the practical test for the hands-on presentation strategy, high computer science aptitude group was the same as the average score for the hands-on presentation strategy, low computer science aptitude group. Within the hands-on presentation strategy group, aptitude was not expected to affect the outcome of the training method. Thus support was provided for research hypothesis H5i.2 However, average achievement scores for the visual presentation strategy, high computer science aptitude group and the verbal presentation strategy, high computer science aptitude group were not greater than the average achievement scores for the visual presentation strategy, low computer science aptitude group and the verbal presentation strategy, low computer science aptitude group. In the visual and verbal presentation strategy groups, aptitude was expected to have an influence on the outcomes of the training methods. The differences were not significant so research hypothesis H5i.3 was not supported. The average score on the practical test for the verbal, high aptitude group was greater than both the verbal, low aptitude group and the visual, low aptitude group, but the visual, high aptitude group had the lowest average score among all the groups. One can only speculate that the visual, high aptitude group may have been bored with the visual presentation strategy, overconfident in their computer science aptitude, and severely frustrated as they could not perform well on the practical test. And as the practical test went on, the dynamic effects of stress could have decreased their performance. The visual, high aptitude group did have the highest increase between pre-treatment computer anxiety and post-treatment computer anxiety than any of the other groups.

5.3. Findings

In the past, researchers have devoted much attention to the measurement of computer anxiety, what form it takes, and the search for correlates of computer anxiety. However, there is a general absence of research on the treatment of computer anxiety. More recently, researchers have begun to investigate training methods as a potential factor for lessening or alleviating computer anxiety. Harrington, McElroy, and Morrow (1990) suggested that training on and exposure to computers with minimal personal discomfort and maximum proficiency should be the goal in the study of computer anxiety. The inevitability of computers and the need for effective achievement of computer skills is no longer an issue for debate. This study attempts to pursue that goal and contribute to the knowledge base for effective training strategies of computer skills and treatment of computer anxiety.

The major findings of this study include:

1. Different presentation strategies of computer skills have very little effect on computer anxiety over a one week investigative period.

2. Hands-on presentation strategy was superior to visual presentation strategy in facilitating student achievement for the use of computer skills when measured by a practical test.

3. Computer science aptitude did not interact with presentation strategies for the training of computer skills.

5.4 Correlation Analysis

Since computer science aptitude was found to be an insignificant factor in this study but other studies (Dwyer, 1978; Konvalina, et al., 1983; Marcoulides, 1988; and McCormick and Ross, 1990) reported that aptitude was one of the most important individual difference variables for the study of learning outcomes, a correlation analysis was performed to identify relationships between aptitude and the other variables investigated in this study. The analysis was performed using Pearson product-moment correlations and significant probabilities reported by SAS. Primarily, it was included to measure the association between the raw data scores from the computer science aptitude test and other continuous variables such as written and practical achievement tests and pre-treatment and post-treatment computer anxiety. Correlation measures the strength of the linear relationship between two variables. If one variable (computer science aptitude) can be expressed exactly as a linear function of another variable (post-treatment computer anxiety), then the correlation is 1 if the variables are directly related or -1 if the variables are inversely related. A correlation of 0 between two variables means that each variable has no linear predictive ability for the other.

The mean score for computer science aptitude was 17.9 for the 81 students involved in this study. The standard deviation for computer science aptitude was 3, a very narrow dispersion for the sample. The mean scores of the high aptitude group (n=27) and of the low aptitude group (n=54) were 21 and 16.4, respectively. These results indicated that the total mean score was only one point from the separation point that divided the groups into high and low computer science aptitude. That is, if a student's computer science aptitude test score was between 20 and 25, then that student was classified as belonging to the high computer science aptitude group. If the student's test score was 19 or below, then that student was classified as belonging to the low computer science aptitude group. The very small variation of these scores among a homogeneous group of engineering students leads to one suspicion why aptitude was not a significant factor in this study.

Another suspicion is the near zero correlation (0.033) between computer science aptitude scores and the practical achievement test scores reported in Table 5.3. Raw scores from the computer science aptitude test did show negative correlations with pre-treatment computer anxiety scores and post-treatment computer anxiety scores. Moderate positive correlations existed between computer science aptitude test scores and written achievement test scores.

Pearson product-moment correlations between the scores for each subject on the pre-treatment computer anxiety questionnaire and their scores on the post-treatment computer anxiety questionnaire were also calculated. The high positive correlation between the pre-treatment computer anxiety scores and the post-treatment computer anxiety scores indicated the pre-treatment computer anxiety variable would be an appropriate covariate in the analysis of covariance on the dependent variable post-treatment computer anxiety. Moderate negative correlations existed between pre-treatment computer anxiety and the other two dependent variables representing the written achievement test and the practical test. Finally, there existed a linear relationship between practical test scores and final course grades for the participants of the study. Table 5.3 presents these correlations.

Table 5.3 Correlation

Correlation Analysis

Simple Statistics

Variable	N	Mean	Std Dev	Sum	Minimum	Maximum
PSAPT	81	17.9383	3.0551	1453.0	7.0000	24.0000
ACHIEVE	81	34.3827	6.8967	2785.0	17.0000	48.0000
PRACT	81	30.2469	17.1738	2450.0	0.0000	50.0000
PSTATE	81	40.1975	12.0181	3256.0	20.0000	70.0000
STATEP	81	45.4074	12.7336	3678.0	22.0000	80.0000
COURSE	81	84.3210	13.3574	6830.0	33.0000	99.0000

PSAPT = computer science aptitude test
ACHIEVE = written achievement test
PRACT = practical test
PSTATE = pre-treatment computer anxiety
STATEP = post-treatment computer anxiety
COURSE = final grade for computer science 201 course

Correlation Analysis

Pearson Correlation Coefficients / Prob > |R| under Ho: Rho=0 / N = 81

	PSAPT	ACHIEVE	PRACT	PSTATE	STATEP	COURSE
PSAPT	1.00000	0.25386	0.03365	-0.22640	-0.20210	0.11076
	0.0	0.0222	0.7655	0.0421	0.0704	0.3249
ACHIEVE	0.25386	1.00000	0.15433	-0.25534	-0.27395	0.20734
	0.0222	0.0	0.1689	0.0214	0.0133	0.0633
PRACT	0.03365	0.15433	1.00000	-0.22917	-0.45832	0.43884
	0.7655	0.1689	0.0	0.0396	0.0001	0.0001
PSTATE	-0.22640	-0.25534	-0.22917	1.00000	0.72194	-0.02283
	0.0421	0.0214	0.0396	0.0	0.0001	0.8397
STATEP	-0.20210	-0.27395	-0.45832	0.72194	1.00000	-0.22206
	0.0704	0.0133	0.0001	0.0001	0.0	0.0463
COURSE	0.11076	0.20734	0.43884	-0.02283	-0.22206	1.00000
	0.3249	0.0633	0.0001	0.8397	0.0463	0.0

5.5. Conclusions

Wells and Bitter (1982) expressed concerns over the frustrations experienced by students with the introduction of computer skills necessary for BASIC programming. Henry and Holtan (1987) stated that no specific design was identified as being the best for computer training of novices. This research has contributed to the knowledge base in the domain of training methods of computer skills for introductory, technical computer courses.

One of the conclusions summarized in section 5.3 of this study indicates that the hands-on presentation strategy, if designed properly, can be more effective than the visual presentation strategy for achieving the use of computer skills when measured by a practical test. Based on a lesson design constructed from task analysis of an instructional system's design model, the training of computer skills was systematically subdivided into a successful, step-by-step approach that defined a series of tasks. Each was a step on the way to the final goal. By successfully completing these successive approximations the student gains satisfaction and acquires patterns that, reshaped at each step, fit into the final whole.

Secondly, the instructor-led presentation provided the students with an enthusiastic leader in order to minimize the discomfort of novices and pace the action-reaction interface between the student and the computer. Third, and most importantly, the first rule in training for a skill was applied: that is, make available the cues that are later to be used in applying the skill. What may seem obvious to some, but all too often, other presentation strategies that do not allow for hands-on training of computer skills may have omitted crucial cues that allow for effective learning strategies.

5.6. Limitations

One limitation of this study is the fact that only group presentation strategies are considered. Other training alternatives that are non-group related, such as individualized, self-paced tutorials, should be considered for future studies.

A second limitation of this study is the use of intact groups that composed the subjects of the three presentation strategies. The three treatment groups consisting of verbal, visual, and hands-on presentation strategies were created by the university's course registration process. Each of the 3 sections of the Computer Science 201 course were intact at the outset of the investigation. There was no random assignment of subjects to treatments.

The statistics that were used in this study comply with internal validity of the experiment when random assignment of intact classrooms to treatments were involved. Campbell and Stanley (1963, p23) suggested "where intact classes have been assigned to treatments, the class means are used as the basic observations, and treatment effects are tested against variations in these means." Mean scores from verbal, visual, and hands-on presentation strategy groups were investigated in this experiment.

Measuring practical experiences at the computer and the amount of time dedicated to practicing computer skills were other limitations. Engineering students have access to personal computers both during controlled training sessions in laboratories and during open laboratory sessions. Some engineering students possess their own personal computers which allow for private hands-on use whenever they desire. Students with personal computers or with open laboratory access to computers have a greater potential for more hands-on practical experiences than students without pc's or open access and thus make it difficult to assess practice time and hands-on experience when investigating the achievement of computer skills and computer anxiety under controlled treatments in a research study.

During this study, the amount of time that the students practiced at the computer was monitored by the instructor. A time-rationing program recorded when a student logged-on to the mainframe computer system and when he/she logged-off. Between the treatments and the practical test, there were six students that accessed the computer system outside of the controlled access time for the experiment. Since the rationing program reported only total login time for each student, there was no way to monitor what facilities were being used on the computer system. Therefore, outside of treatment computer access was not restricted so additional practice time may have occurred. However, when the six students were questioned about their login times between the laboratory computer exercises and the practical test, they indicated that they were checking their electronic mail and not practicing programming skills. Further investigation indicated that the six students were from the verbal group and their combined mean practical test score was 31, which was 1 point different than the mean practical test score of 30 for the verbal presentation strategy group. This was not significantly different from expectations that no impact was encountered from additional computer time.

Finally, only computer skills that are necessary for programming a computer were investigated in this study. The skills or abilities necessary to use computers equipped with preprogrammed software (such as word processing, spreadsheets, data bases, and electronic mail) are not the same as the skills or abilities necessary to understand the operations of computers and to program them. Future studies should take into consideration the training of all types of computer use. Further exploration of different presentations strategies could be incorporated in future studies to investigate efficacy and anxiety levels associated with technical, computer programming courses and non-technical (that is, non-programming) computer appreciation courses.

5.7. Recommendations

The findings and conclusions of this study provide several possible future research topics. This study indicates that hands-on presentation strategy may serve as reinforcement supporting the realism continuum and Cronbach's research on the use of realism for the learning of skills.

To explore further the use of realism in the training of computer skills, self-paced, individualized, hands-on instruction for students with low computer anxiety could be compared with instructor-led, group-presented, hands-on instruction for students with high computer anxiety. The fact that hands-on presentation strategy affected the achievement of computer skills suggests that further consideration of environmental factors, particularly those involving specific features of training programs, along with dispositional characteristics of computer anxious individuals, may prove informative.

Future consideration should be given to extending the treatment methods for more than a week to allow possible changes to occur over a longer period of time. Then investigations could be conducted to determine the effectiveness of training methods and changes of computer anxiety levels over the entire duration of the course. It would be of interest to record computer anxiety levels at all phases covering the objectives of the course, possibly spanning ten or fifteen weeks. These phases (reported by

Fitts and Posner, 1976) include the early or cognitive phase (novice tries to "understand" the task and what it demands), the intermediate or associative phase (which lasts for varying periods of time depending on the complexity of the skill and the extent of the practice required for new integrations), and the final or autonomous phase (in which skills require less processing and are less subject to interference from environmental distractions). Computer anxiety levels could be measured at monthly intervals to assess possible changes that may occur from learning curves, where performance is considered a function of days or months of practice under a particular presentation strategy.

REFERENCES

Banks, Mark J. & Havice Michael J., (1989). Strategies for dealing with computer anxiety: two case studies. *Educational Technology*, 29(1), 22-26.

Bikson, T. K. and Gutek, B. A., (1983). Training in Automated Offices: An Empirical Study of Design and Methods, in Training for Tomorrow, *Proceedings of IFAC/IFIP, 1983 Symposium*, (pp. 129-143).

Briggs, L. J. and Wager, W. W., (1981). *Handbook of procedures for the design of instruction*. Englewood Cliffs, NJ: Educational Technology Publications, Inc.

Cambre, J. and Cook, D., (1984). Computer Anxiety: Definition, measurement, and, correlates. Paper presented at the *Annual Meeting of the American Research Association*. ERIC Document ED 246 085.

Campbell, D. T. and Stanley, J. C., (1963). *Experimental and Quasi-Experimental Designs for Research*. Boston, MA: Houghton Mifflin Company.

Cattell, R. B., (1966). *Patterns of Change: Measurement in relation to state dimension, trait change, ability, and process concepts*. Handbook of Multivariate Experimental Psychology. Chicago, IL.: Rand McNally.

Cattell, R. B. and Scheier, I. H., (1961). *The Meaning of Measurement of Neuroticism and Anxiety*. New York, NY.: Ronald Press.

Chen, M., (1986). Gender and Computers: The Beneficial Effects of Experience on Attitudes. *Journal of Educational Computing Research*, 2(3), 265-282.

Cronbach, Lee J., Snow, Richard E., (1977). *Aptitudes and Instructional Methods*. New York, NY.: Irvington.

Cushall, M. B., (1989). Research on Computer Anxiety: A Review of the Literature and Suggestions for Future Research Activities. *31st ADCIS Conference Proceedings,* (pp237-240) Crystal City, VA.

Dale. E., (1954). *Audiovisual Methods in Teaching.* New York: Holt.

Dwyer, Carol A., (1984). *The Effect of Varied Rehearsal Strategies in Facilitating Achievement of Different Educational Objectives as Measured by Verbal and Visual Testing Modes.* Unpublished doctoral dissertation, The Pennsylvania State University.

Dwyer, Carol A., (1986). The effect of rehearsal method and rehearsal time on students' information acquisition and retrieval. *Journal of Instructional Psychology,* 13(1), 25-31.

Dwyer, Carol A., (1986). The effect of varied rehearsal strategies in facilitating achievement of different educational objectives measured by verbal and visual testing models. *Journal of Experimental Education,* 54(2), 73-84.

Dwyer, Carol A. & Dwyer, Francis M., (1987). Effect of depth of information processing on students' ability to acquire and retrieve information related to different instructional objectives. *Programmed Learning and Educational Technology,* 24(4), 264-279.

Dwyer, Francis M., (1978). *Strategies for Improving Visual Learning.* State College, PA.: Learning Services.

Dwyer, Francis M., (1985). The effect of visualized instruction and varied rehearsal and evaluation strategies (verbal and visual) in facilitating students' long-term retention on test measuring different instructional objectives. *Journal of Visual/Verbal Languaging*, 5(2), 15-27.

Ferguson, G. A., (1981). *Statistical Analysis in Psychology and Education.* 5th Edition. New York: McGraw-Hill.

Finn, J. D., (1953). Professionalizing the audio-visual field. *Audio-Visual Communications Review*, 1, 6-17.

Fitts, P. M., and Posner, M. I. (1967). *Human performance.* Monterey, CA: Brooks/Cole.

Gagne, Ellen D., (1985). *The Cognitive Psychology of School Learning.* Boston and Toronto.: Little, Brown & Company.

Gagne, Robert M., (1977). *The Conditions of Learning.* 3rd Edition. New York, NY.: Holt. Rinehart and Winston.

Glynn, S. M., and DiVesta, F. J., (1977). Outline and hierarchical organization as aids for study and retrieval. *Journal of Educational Psychology*, 69, 89.

Harrington, Kermith V., McElroy, James C., Marrow, Paula C., (1990). Computer anxiety and computer-based training: a laboratory experiment. *Journal of Educational Computing Research*, 6(3), 343-358.

Hartman, F. R., (1961). Recognition learning under multiple channel presentation and testing conditions. *Audio Video Communications Review*, 9, 24-43.

Henry, C. M. and Holtan, B. D., (1987). Introducing Novices to the Computer: Hands-on vs. Demonstration. *School Science and Mathematics*, 87(7), 598-607.

Honeyman, D. and White, W., (1987). Computer anxiety in educators learning to use the computer: A preliminary report. *Journal of Research on Computing in Education*, 20(2), 129-137.

Hortin, J. A., and Bailey, G. D. (1982). Mental Rehearsal and Visual Literacy. *Journal of Instructional Psychology*, 9, 202-213.

Howard, G. S. (1986). *Computer Anxiety and Management Use of Microcomputers*, Ann Arbor, MI.: UMI Research Press.

Howard, G. S., (1986). *Computer Anxiety and Other Determinants of Managers' Attitudes Toward the Usefulness of Microcomputers in Management.* Unpublished doctoral dissertation, Kent State University.

Howard, G. S., Murphy, C. M., and Thomas, G. N., (1987). Computer anxiety considerations for design of introductory computer courses. *Educational Research Quarterly*, 11(4), 12-22.

Hsia, H. J., (1971). The Information Processing Capacity of Modality and Channel Performance. *Audio Video Communications Review*, 19,51-75.

Jay, G. M. and Willis, S. L., (1992). Influence of direct computer experience on older adults' attitudes toward computers. *Journal of Gerontology: PSYCHOLOGICAL SCIENCES*, Vol. 47, No. 4, p250-p257.

Jonassen, D., (1986). State Anxiety and Exposure to Microcomputers: Assessing Computerphobia. Paper presented at the *Annual Convention of the Association for Educational Communications and Technology*, Las Vagas, NV., 8 P., ERIC Document ED 267-779.

Jones, P., Wall, R., (1985). *Computer experience and computer anxiety: Two pilot studies*. Reports - Research/Technical. ERIC Document ED 275 315.

Joseph, John H., (1987). Field-dependence/independence and the mathemagenic effects of observable visual imagery rehearsal for learning from computer-presented text. *Proceedings of the Association for Educational Communications and Technology*, Atlanta, GA.

Konvalina, J., Stephens, L., and Wileman, S., (1981). Computer Science Aptitude Test. *University of Nebraska at Omaha*.

Konvalina, J., Stephens, L., and Wileman, S., (1983). Math Proficiency: A Key to Success for Computer Science Students. *Communications of the Association for Computing Machinery*, Volume 26, Number 5, pages 377-382.

Levie, W. H., and Levie, D., (1975). Pictorial Memory Processes. *AV Communication Review*, 23, 81-97.

Leso, T. J., Peck, K. L., (1992). Computer Anxiety and Different Types of Computer Courses. *Journal of Educational Computing Research*, 8(4), 469-478.

McKeachie. Wilbert J., (Eds), (1980). *Learning, Cognition and College Teaching. New directions for teaching and learning, No. 2.* San Francisco, CA.: Jossey-Bass, 89.

Marcoulides, G. A., (1988). The Relationship Between Computer Anxiety and Computer Achievement. *Journal of Educational Computing Research*, 4(2), 151-158.

McCormick, D., Ross, S. M., (1990). Effects of Computer Access and Flowcharting on Students' Attitudes and Performance in Learning Computer Programming. *Journal of Educational Computing Research*, 6(2), 203-213.

Morrow, P. C., (1986). Attitudinal and Behavioral Correlates of Computer Anxiety. *Psychological Reports*, 59(3), 1199-1204.

O'Neil, Harold F., Jr., Spielberger, Charles D., (Eds.), (1992). *Cognitive and Affective Learning Strategies*. New York, NY. Academic Press.

Paivio, A., (1971) *Imagery and Verbal Process.* NY: Holt, Rienhart and Winston.

Paul, L., (1982). Research on Cyberphiliacs, Cyberphobiacs Reveals 30% of Workers Fear Computers. *Computerworld,* 15, p. 14, April 5, 1982.

PSU (1989). Bulletin 1989-1990 The Pennsylvania State University Baccalaureate Degree Programs, University Park, PA.

Raub, A. C., (1981). *Correlates of Computer Anxiety in College Students*, unpublished doctoral dissertation, University of Pennsylvania, Philadelphia.

Rickards, J. P., (1980). Notetaking, underlining, inserted questions and organizers in text: Research conclusions and educational implications. *Educational Technology,* 20, 5-11.

Rickards, J. P., and DiVesta, F. J., (1974). Type and frequency of questions in processing textual material. *Journal of Educational Psychology*, 66, 354-362.

Simon, H. A. (1980). Problem solving and education. In D. T. Tuma, F. Reif (Eds.), *Problem solving and education.* Hillsdale, NJ: Erlbaum.

Simonson, M. R., Matt, A., and Maurer, M. M., (1987). Development of a Standardized Test of Computer Literacy and a Computer Anxiety Index. *Journal of Educational Computing Research*, 3(2), 231-247.

Spielberger, C. D., (1966). Theory and research on anxiety. In C. D. Spielberger (Ed.), *Anxiety and Behavior*, New York, NY.: Academic Press.

Spielberger, C. D., (1972). *Anxiety as an emotional state. Anxiety: Current Trends in Theory and Research*, 1. New York, NY.: Academic Press.

Spielberger, C. D., (1976). The Nature and Measurement of Anxiety. *Cross-Cultural Anxiety*. Washington, D.C.: Hemisphere/Wily.

Spielberger, C. D., (1979). *Preliminary Manual for the State-Trait Personality Inventory (STPI)*. University of South Florida.

Spielberger, C. D., (1983). *MANUAL FOR THE STATE-TRAIT ANXIETY INVENTORY (FORM Y)*. Palo Alto, CA.: Consulting Psychologists Press, Inc.

Weinstein, Claire E., & Mayer, Richard E., (1983) The teaching of learning strategies. *Innovation Abstracts*, 5(32).

Wells, M. and Bitter, G., (1982). The first step in utilizing computers in education: Preparing computer literate teachers. Tempe, Arizona: Arizona State University. ERIC Document Reproduction No. 218 703.

Wittrock, M. C., Lumsdaine, Arthur A., (1977) Instructional psychology. *Annual Review of Psychology*, 28, 417-459.

Appendix A

GLOSSARY

ACHIEVE	Written achievement test.
ANOVA	Univariate analysis of variance.
ATI	Aptitude treatment interaction.
BASIC	Beginners all purpose software programming language.
CAI	Computer assisted instruction.
CAIN	Computer anxiety index.
COURSE	Final course grade for students enrolled in computer science 201.
CS 201	Computer Science 201 programming course for engineers.
FORTRAN	Scientific and engineering software programming language.
GLM	General linear model for statistical analysis.
HoH	Hands-on presentation strategy high computer science aptitude.
HoL	Hands-on presentation strategy low computer science aptitude.
H1	Research hypothesis for presentation strategy on written test.
H2	Research hypothesis for presentation strategy on computer anxiety.
H3	Research hypothesis for aptitude on written test.
H4	Research hypothesis for aptitude on computer anxiety.

H5	Research hypothesis for interaction and main effects on practical test.
IBM VM/CMS	International business machine software operating system.
MANOVA	Multivariate analysis of variance.
PSTATE	Pre-treatment computer anxiety.
PRACT	Practical achievement test.
PROC CORR	Software procedure for correlation analysis.
PSAPT	Computer science aptitude raw test scores.
SAS	Statistical analysis system.
STAI	State-trait anxiety inventory.
STATEP	Post-treatment computer anxiety.
TYPE I SS	Sequential sum of squares used for balanced data.
TYPE III SS	Partial sum of squares used for unbalanced data.
VeH	Verbal presentation strategy high computer science aptitude.
VeL	Verbal presentation strategy low computer science aptitude.
ViH	Visual presentation strategy high computer science aptitude.
ViL	Visual presentation strategy low computer science aptitude.
VISICALC	Spreadsheet software application program.
WFOR	Waterloo microFORTRAN software programming system.

Appendix B

COMPUTER SCIENCE APTITUDE TEST

PART I
(Sequences and Logic)

For questions 1 through 4, look for a pattern and fill in the missing term in the sequence:

1. A, C, F, H, K, M, ________
   (a) N (b) O (c) P (d) Q (e) R

2. ABC, ABD, ABE, ACD, ACE, ________
   (a) ADE (b) ACB (c) AED (d) ADC (e) AEC

3. $\frac{1}{4}, \frac{3}{6}, \frac{2}{5}, \frac{4}{7}, \frac{3}{6},$ ________
   (a) $\frac{1}{2}$ (b) $\frac{3}{8}$ (c) $\frac{4}{5}$ (d) $\frac{5}{8}$ (e) $\frac{2}{5}$

4. 1, 1, 2, 3, 5, 8, ________
   (a) 13 (b) 8 (c) 11 (d) 9 (e) 40

5. How many numbers are there in the sequence below if all the missing terms (indicated by ...) are included?

   0, 3, 6, 9, 12, 15, ..., 240

   (a) 240 (b) 241 (c) 80 (d) 81 (e) none of these

6. A teacher said to a student, "If you receive an A on the final exam, then you will pass the course." Suppose the student did not pass the course. What conclusion is valid?
   (a) The student received an A on the final exam.
   (b) The student did not receive an A on the final exam.
   (c) The student flunked the final exam.
   (d) If the student passed the course, then he or she received an A on the final.
   (e) None of these is valid.

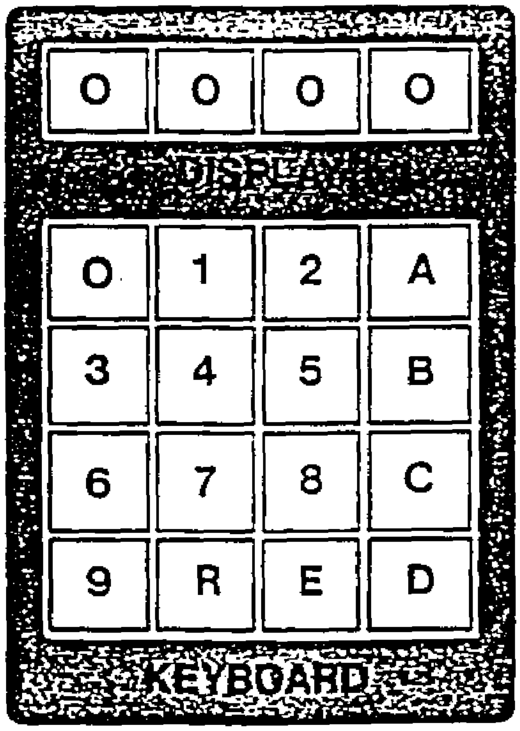
7. John said to Jane, "If it rains, then I won't play tennis." Suppose it did not rain; then what conclusion is valid?
   (a) John played tennis.
   (b) John did not play tennis.
   (c) If John does not play tennis, then it rains.
   (d) It did not rain and John played tennis.
   (e) None of these is valid.

8. Suppose all computers are logical devices, and some computers are bistable. What conclusion is valid?
   (a) All logical devices are bistable.
   (b) All computers are bistable.
   (c) Some computers are not logical devices.
   (d) Some logical devices are computers.
   (e) None of these is valid.

9. Think of a number. Add 3 to the number. Multiply your answer by 2. Subtract 4 from your answer. Divide by 2. Subtract the number with which you started. Your answer is
   (a) 0 (b) 1 (c) 2 (d) negative (e) none of these

10. Which one of the words does not belong to the group?
    (a) REDDER (b) BETTER (c) RADAR
    (d) PEEP (e) POP

PART II
(Calculator Simulator)



Consider the calculator shown here with a four-digit display, digits 0 through 9, and operations A, B, C, D, E, and R. The meaning of the operations of the calculator are as follows:

R = Reset the display so that all digits are 0.

E = Enter the number pressed after the letter E into the display.

A = Add the number pressed after the letter A to the number in the display and display the result (sum).

B = Subtract the number in the display from the number pressed after the letter B and display the result (difference).

C = Multiply the number pressed after the letter C by the number in the display and display the result (product).

D = Divide the number pressed after the letter D into the number in the display and display the result (whole-number quotient).

Note: Except for the letter R, a number is pressed after a letter.

An example of a calculator *program is the following (instructions are performed from left to right in order): RE20B50D6. This program first resets the display to 0, then enters the number 20 into the display, subtracts 20 from 50 (the display now reads 30), and finally divides the result in the display by 6. The display reads 5 after the last operation.*

Now answer the following questions based on the calculator above:

11. A student scored 85, 66, and 92 on three exams. Which calculator program will display the average of the three exams after the last operation?
    - (a) RE85A66B92D3
    - (b) RE85A66D3A92
    - (c) RE92A66A85D3
    - (d) RE92A66A85C3
    - (e) none of these

12. Mrs. Gross bought four single grocery items at the following prices: 69 cents, 45 cents, 12 cents, 37 cents. Also, she bought 5 pounds of bananas at 39 cents a pound. She paid for the groceries with a $10 bill (1,000 cents): Which calculator program will display her correct change (in cents)?
    - (a) RE69A45A12A39B1000
    - (b) RE39C5A69A45A12B1000
    - (c) RE69A45A12A37A39C5B1000
    - (d) RE5C39A69A45A12A37B1000
    - (e) none of these

13. Find the number displayed after the last operation of the following calculator program:

    RE6A0B8C2C4D16B1

    (a) 2 (b) 0 (c) 1 (d) a negative number (e) none of these

14. Which statement best describes the following calculator program?

    RE13D3C3B13

    - (a) divides two numbers, 13 and 3, and displays the quotient
    - (b) divides, multiplies the result, and finally displays 13 again
    - (c) computes and finally displays a negative number
    - (d) computes and displays 3 times 13 minus 13
    - (e) computes and displays the remainder when 13 is divided by 3

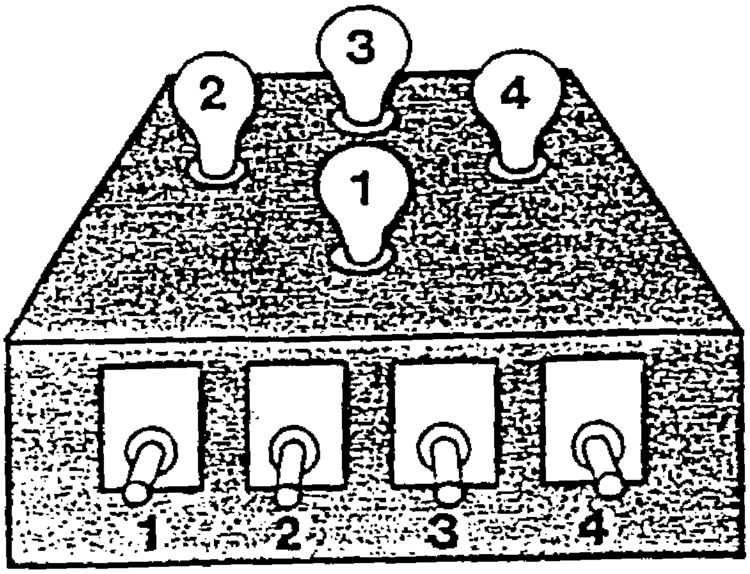
15. What last operation must be added to the following program so that the display will read 1 after the last operation?

    RE12A3C2B56D2 ________

    (a) D14 (b) A1 (c) B13 (d) B12 (e) none of these

PART III
(Algorithm)

Assume we have four light bulbs arranged in a circle labeled 1, 2, 3, and 4 as shown in the figure.



Assume further that we have four switches connected so that each switch controls the light bulb with the corresponding number. Consider the following set of instructions, but do not perform the actions indicated yet.

1. *Turn on the light bulb that is directly across from the single light bulb that is on.*
2. *If any odd-numbered light bulb is on, go to step 4.*
3. *Turn off the lowest-numbered light bulb, and go to step 5.*
4. *Turn off the highest-numbered light bulb.*
5. *Turn on the bulb next to the highest-numbered bulb that is on, in a clockwise direction.*
6. *Turn off any even-numbered bulbs which might be on, and stop.*

Now answer the following questions:

16. Assume only light bulb 1 is on. Carry out the instructions, starting with step 1. When you stop in step 6,
    - (a) light bulbs 3 and 4 are on
    - (b) no light bulbs are on
    - (c) only light bulb 1 is on
    - (d) only light bulb 2 is on
    - (e) none of the above

17. Carry out the instructions again. This time assume only light bulb 2 is on at the beginning. When you stop in step 6,
    - (a) only light bulb 1 is on
    - (b) light bulbs 2 and 3 are on
    - (c) at least three light bulbs are on
    - (d) only two light bulbs are on
    - (e) none of the above

18. Again carry out the instructions, this time assuming only light bulb 3 is on initially. When you stop in step 6,

(a) only bulb 2 is on
(b) only bulb 3 is on
(c) only bulb 4 is on
(d) all bulbs are on
(e) none of the above

19. Finally, carry out the instructions assuming only bulb 4 is initially on. When you stop in step 6,

(a) light bulbs 2 and 4 are on
(b) light bulbs 1 and 3 are on
(c) at least one even-numbered bulb is on
(d) at least one odd-numbered bulb is on
(e) none of the above

20. Based on your experience in carrying out the instructions,

(a) the instructions can be applied regardless of the number of light bulbs initially turned on
(b) regardless of which light bulb is initially on, when we stop in step 6 all light bulbs will be off
(c) regardless of which light bulb is initially on, when we stop in step 6 only light bulb 1 will be on
(d) when an even-numbered bulb is initially turned on, then when we stop in step 6 only light bulb 3 will be on
(e) none of the above

PART IV
(Word Problems)

21. Six times a number is 3 more than twice the number. What is the number?

(a) $4/3$ (b) $-3/4$ (c) $3/4$ (d) $-4\frac{1}{2}$
(e) none of these

22. A bank contains nickels and dimes. The total value of the coins is $2.10 and there are three more dimes than nickels. How many nickels are there?

(a) 13 (b) 12 (c) 15 (d) 36 (e) none of these

23. A law requires that the amount of chicken used in hot dogs cannot exceed 25 percent of the total weight of the hot dog. How many ounces could a hot dog weigh if it contained 1.5 ounces of chicken?

(a) 3 (b) 5 (c) 5.5 (d) 6 (e) none of these

24. A farmer mixes seed worth 15 cents per pound with seed worth 20 cents per pound to produce a mixture of 50 pounds of seed worth 18 cents per pound. How many pounds of seed worth 20 cents per pound did he use in the mixture?

(a) 20 (b) 25 (c) 30 (d) 40 (e) none of these

25. Volumes 12 through 29 of an encyclopedia have misprints on pages 21 through 53 of each volume. How many pages in the encyclopedia have misprints?

(a) 18 (b) 32 (c) 544 (d) 594 (e) none of these

Appendix C

PRE AND POST TREATMENT QUESTIONNAIRE

FOR STATE COMPUTER ANXIETY

Appendix D

WRITTEN ACHIEVEMENT TEST

AFTER TREATMENT

Prerequisite Skills To Prepare Your Programming Assignments Exam

Answers to questions 1 - 6 are worth 50 points, approximately 1 each.

1. State a definition for a computer programming editor (such as WFOR). Include at least two functions of the editor.

____

____

____

2. List the sequence for "CREATING AND SAVING A NEW FILE," (PSU FVM System using PS2), by writing the numbers 1 - 5 in front of the steps (a) - (e) below.

_______ (a) Type I and depress ENTER key

_______ (b) Type WFOR and depress ENTER key

_______ (c) Type PUT filename and depress ENTER key

_______ (d) Depress F9

_______ (e) Type new lines and depress ENTER key after each line

3. List the sequence for "RUNNING YOUR PROGRAM AND MAKING A LISTING" of output generated during a WFOR program's RUN, by writing the numbers 1 - 9 in front of the steps (a) - (i) below.

_______ (a) Type GET PROG1 and depres ENTER key

_______ (b) Type LISTOFF CONSOLE and depress ENTER key

_______ (c) Type LOGOFF and depress ENTER key

_______ (d) Type DIR and depress ENTER key

_______ (e) Type TYPE PROG1 FILE A1 and depress ENTER key

_______ (f) Type RUN and depress ENTER key

_______ (g) Type WFOR and depress ENTER Key

_______ (h) Type *D and depress ENTER Key

_______ (i) Type BYE and depress ENTER Key

-2-

4. On the blank line preceding each command (a) - (p) for the PSU Computer System, write the C and/or W or N of the matching type of command for CMS, WFOR, or Not Applicable.

_____ (a) I	_____ (e) GET	_____ (i) WFOR	_____ (m) .,+2
_____ (b) DIR	_____ (f) TYPE	_____ (j) LISTOFF	_____ (n) 9
_____ (c) SAVE	_____ (g) PUT	_____ (k) CH	_____ (o) $
_____ (d) *D	_____ (h) BEGIN	_____ (l) RUN	_____ (p) BYE

5. Following each description (a) - (g) below of editing a file using WFOR, circle, INSERT, CHANGE, and/or DELETE to classify the type of editing capability.

(a) Depress keypad 3; depress up arrow key; move cursor and retype; depress keypad 3.

INSERT CHANGE DELETE

(b) Depress keypad 3; move cursor to directly above the line; depress keypad 5; depress ENTER key.

INSERT CHANGE DELETE

(c) Depress keypad 3; position cursor to the line; depress keypad 6; depress keypad 3.

INSERT CHANGE DELETE

(d) Depress keypad 3; move cursor to line; depress keypad 3; type CH/before/after and depress ENTER key.

INSERT CHANGE DELETE

(e) Depress keypad 3; move cursor to directly above the line; depress keypad 3; Type I and depress ENTER key.

INSERT CHANGE DELETE

(f) Depress keypad 3; move cursor to line; depress keypad 3; type .,+ 2 D and depress ENTER key.

INSERT CHANGE DELETE

(g) Type *D and depress ENTER key; Type I and depress ENTER key.

INSERT CHANGE DELETE

6. Given the following five lines in WFOR, edit by writing the correct FORTRAN syntax (i.e., correct all errors by rewriting the five lines).

```
(THIS IS OUR FIRST FORTRAN EXAMPLE)
PROGRAM EXAMPLE 1 BY LESO
STOP
PRINT, *THIS IS MY FIRST FORTRAN PROGRAM*
END.
```

____

____

____

____

____

Appendix E

EXERCISE BEFORE PRACTICAL TEST

WFOR EDITOR EXERCISES

Answers 1 - 7 Record Below
Answers 8 - 12 Computer Listings

Circle Yes or No — 1. Investigate the editor (WFOR) on the system PSUVM you will be using. Which of the following operations can it do?

- Y N — a. Reverse two adjacent characters (very useful because it fixes a common typing mistake).
- Y N — b. Include the contents of some other file inside the file you are editing.
- Y N — c. Search for a particular word or phrase.
- Y N — d. Show a particular line of the file (say line 56).

______, ______ — 2. What languages (other than Fortran) are available on your system?

Circle W or M — 3. Is there word-processing software available on your system? Is it wysiwyg or markup system?

Modified→ — 4. Type in the sample program from LESO'S GUIDE* 3.1, using your text editor. Compile the program and execute it. Do you get the same output? Don't be surprised if you get some error messages from the compiler at first. It's hard to type programs without making mistakes. Your system may require that all the characters be entered in uppercase (capitals).

Record Output: ______________

Write Answers: ____________ ____

Yes — 1 Meg Byte — 5. What command do you use to list your files on the computer system you work with? Is the file system of this computer system hierarchically structured? Is there a limit to the amount of space available to you? If so, what is it.

Yes — 6. Does your computer system have a debugger for use with Fortran Programs? If so, obtain documentation so that you can use it in the remainder of this course. Handout later.

7. Intentionally omitted.

Modified→ — 8. Use your editor and compiler to enter and execute the following program. Type carefully to avoid diagnostic messages. Follow Leso's Guide 6.3 for a Hard Copy of the assignments 8 - 12. Bring to class on Tuesday, 5 listings of output generated during a program's run.

```
program first
integer i, j
i = 10
j = 2
print *, i/j, i, j
stop
end
```

*Consolidate 5 listings into 1 by issuing Type & Run 5 times during Listoff Console.

9. Use your editor to modify the program in Exercise 8 by replacing the word "print" with the word "priny" (which could happen if you pressed the wrong key). Rerun the program and make Hard Copy.

10. Use your editor to modify the program in Exercise 8 by replacing the number 2 with the number 0. Rerun the program and make Hard Copy

11. Use your editor to modify the program in Exercise 8 by replacing the number 10 with the number 9. Rerun the program and make a Hard Copy. This will be explained in a later chapter.

12. Use your editor to modify the program in Exercise 11 by replacing the word integer with the word real. Rerun the program and make Hard Copy. This will be explained in a later chapter.

NOTE: For 9 - 11 Always Modify the Program in Exercise 8

* Reference: THE ALTOONA CAMPUS (CLRC) IBM USER'S GUIDE, by T. J. Leso, here after referred to as LESO'S GUIDE. (See Sections 3.1 p.7 and 6.3 p. 14).

Appendix F

PRACTICAL ACHIEVEMENT TEST

ONE-WEEK AFTER TREATMENT

```
Path: psuvm!tll
Organization: Penn State University
Date: Thu, 10 Feb 1994 09:39:25 EST
From: Tim Leso  Penn State Altoona <T1L@psuvm.psu.edu>
Message-ID: <94041.093925T1L@psuvm.psu.edu>
Newsgroups: private.cs201.1.2
Subject: Error1 file a
Lines: 22

****
****  Practical Achievement Test--ERROR1
****  Parts 1 and 4 out of 5 parts.
****  Part 1:
****  Retrieve this Fortran file which contains one syntax error.
****  Correct the error and save the file as ERROR1 FILE A1.
****  Print the corrected file ERROR1 FILE A1 for 10 points.
****  Part 4:
****  Retrieve the corrected file ERROR1 FILE A1.
****  Execute and print the correct output for 10 points.
****  Refer to ERROR2 for Parts 2, 3, and 5 of the Test.
****
program separate
interger wholepart
real fractionpart, realnumber
print, "please enter real number, for example 77.61 "
read, realnumber
wholepart = int(realnumber)
fractionpart = realnumber - wholepart
print, wholepart, fractionpart, " are parts of ", realnumber
stop
end
```

```
Path: psuvm!til
Organization: Penn State University
Date: Thu, 10 Feb 1994 09:40:00 EST
From: Tim Leso  Penn State Altoona <TIL@psuvm.psu.edu>
Message-ID: <94041.094000TIL@psuvm.psu.edu>
Newsgroups: private.cs201.1.2
Subject: error2 file a
Lines: 34

****
**** Practical Achievement Test--ERROR2
**** Parts 2, 3, and 5 out of 5 parts.
**** Part 2:
**** Retrieve this Fortran file which contains one execution error.
**** Debug this error by running the program and detect and
**** correct the instruction in error.
**** Save the corrected file as ERROR2 FILE A1.
**** Print the corrected file ERROR2 FILE A1 for 10 points.
**** Part 3:
**** Print a directory of the files contained on your
**** "A" Disk, including ERROR1 and ERROR2, for 10 points.
**** Part 5:
**** Retrieve the corrected file ERROR2 FILE A1.
**** Execute and print the correct output for 10 points.
**** Refer to ERROR1 for Parts 1 and 4 of the Test.
****
program fraction
integer power, leftdigit, numdigits, base
real sum10fraction, fractionpart
base = 2
numdigits = 3
fractionpart = 0.101
print, "fraction entered: ",fractionpart," with digits ",numdigits
sum10fraction = 0.0
do power = -1, -numdigits, -1
  leftdigit = int(fractionpart * 10)
  fractionpart = fractionpart * 10 - leftdigit
  fractionpart = fractionpart + 1.0 * 10.0**(-numdigits)
  sum10fraction = sum10fraction + leftdigit * base**power
enddo
print, "base 10 of base 2 number 0.101 is: ",sum10fraction
stop
end
```

```
1type error1 file a1

****
**** Practical Achievement Test--ERROR1
**** Parts 1 and 4 out of 5 parts.
**** Part 1:
**** Retrieve this Fortran file which contains one syntax error.
**** Correct the error and save the file as ERROR1 FILE A1.
**** Print the corrected file ERROR1 FILE A1 for 10 points.
**** Part 4:
**** Retrieve the corrected file ERROR1 FILE A1.
**** Execute and print the correct output for 10 points.
**** Refer to ERROR2 for Parts 2, 3, and 5 of the Test.
****
program separate
integer wholepart
real fractionpart, realnumber
print, "please enter real number, for example 77.61 "
read, realnumber
wholepart = int(realnumber)
fractionpart = realnumber - wholepart
print, wholepart, fractionpart, " are parts of ", realnumber
stop
end

Executing...
please enter real number, for example 77.61
77.61
         77  .609999999999999 are parts of   77.6100000000000
...Stop

Ready; T=0.00/0.01 09:45:05
1type error2 file a1

****
**** Practical Achievement Test--ERROR2
**** Parts 2, 3, and 5 out of 5 parts.
**** Part 2:
**** Retrieve this Fortran file which contains one execution error.
**** Debug this error by running the program and detect and
**** correct the instruction in error.
**** Save the corrected file as ERROR2 FILE A1.
**** Print the corrected file ERROR2 FILE A1 for 10 points.
**** Part 3:
**** Print a directory of the files contained on your
**** "A" Disk, including ERROR1 and ERROR2, for 10 points.
**** Part 5:
**** Retrieve the corrected file ERROR2 FILE A1.
**** Execute and print the correct output for 10 points.
**** Refer to ERROR1 for Parts 1 and 4 of the Test.
****
program fraction
integer power, leftdigit, numdigits, base
real sum10fraction, fractionpart
base = 2
```

```
numdigits = 3
fractionpart = 0.101
print, "fraction entered: ",fractionpart," with digits ",numdigits
sum10fraction = 0.0
do power = -1, -numdigits, -1
  leftdigit = int(fractionpart * 10)
  fractionpart = fractionpart * 10 - leftdigit
  fractionpart = fractionpart + 1.0 * 10.0**(-numdigits)
  sum10fraction = sum10fraction + leftdigit * base**power
enddo
print, "base 10 of base 2 number 0.101 is: ",sum10fraction
stop
end

Ready; T=0.00/0.01 09:45:30
vfor
ERROR1E  FILE    A1
ERROR2E  FILE    A1
ERROR2X  FILE    A1
ERROR1   FILE    A1
ERROR2   FILE    A1

Executing...
fraction entered:   .101000000000000 with digits          3
base 10 of base 2 number 0.101 is:   0.00000000000000
...Stop
Ready; T=0.02/0.05 09:47:34

LOGOFF
```

Appendix G

SCRIPT FOR TREATMENTS (HAND-OUT) AKA LESO'S GUIDE

THE ALTOONA CAMPUS COMPUTER AND LEARNING RESOURCES CENTER (CLRC) IBM USER'S GUIDE

---

by

T. J. Leso

IBM Virtual Machine/370 Conversation Monitor System (VM/370 CMS)

and

University of Waterloo Editor (WEDIT), Interpreter Fortran (WFOR),

plus An Overview of Electronic Mail (E-Mail)

with on-line, interactive, full-screen IBM Personal System/2 Model 50Z and Model 70 Terminals

Version 12 January 1992

TABLE OF CONTENTS

ACKNOWLEDGMENT

The Author thanks Suzanne Kenawell for her superior talent to manipulate the many changes and versions of this Guide to remain current with the PSU VM/CMS environment, and the CAC/CES revisions.

1.0 OVERVIEW

The Altoona Campus IBM User's Guide is intended to be used by novices who are taking an introductory Computer Science course. The User's Guide will demonstrate how to communicate with the IBM computer, how to prepare your programming assignments from the Video Display Terminal, and how to secure paper copy of your assignments which will be stored on computer disk files. The User's Guide is specifically tailored for the students of the Altoona Campus. Although other university guides and manuals exist on these how to's, this User's Guide is the most concise documentation for your course at Altoona. Nevertheless, a list of references is included for additional information. You can find these references in the University Park Computation Center Library.

2.0 SO YOU'RE TAKING A COMP SCI COURSE

There will be three tasks that you need to do for your COMP SCI course from the computer laboratory.

TASK 1. Communicate with the computer.
TASK 2. Prepare your programming assignments.
TASK 3. Secure paper copies of your assignments.

We will cover these tasks separately and in the above logical order. Task one is generally called "LOGGING ON TO THE COMPUTER". Task two is "CREATING, EDITING, AND SAVING YOUR FILE". Task three is "RUNNING YOUR PROGRAM AND MAKING A LISTING".

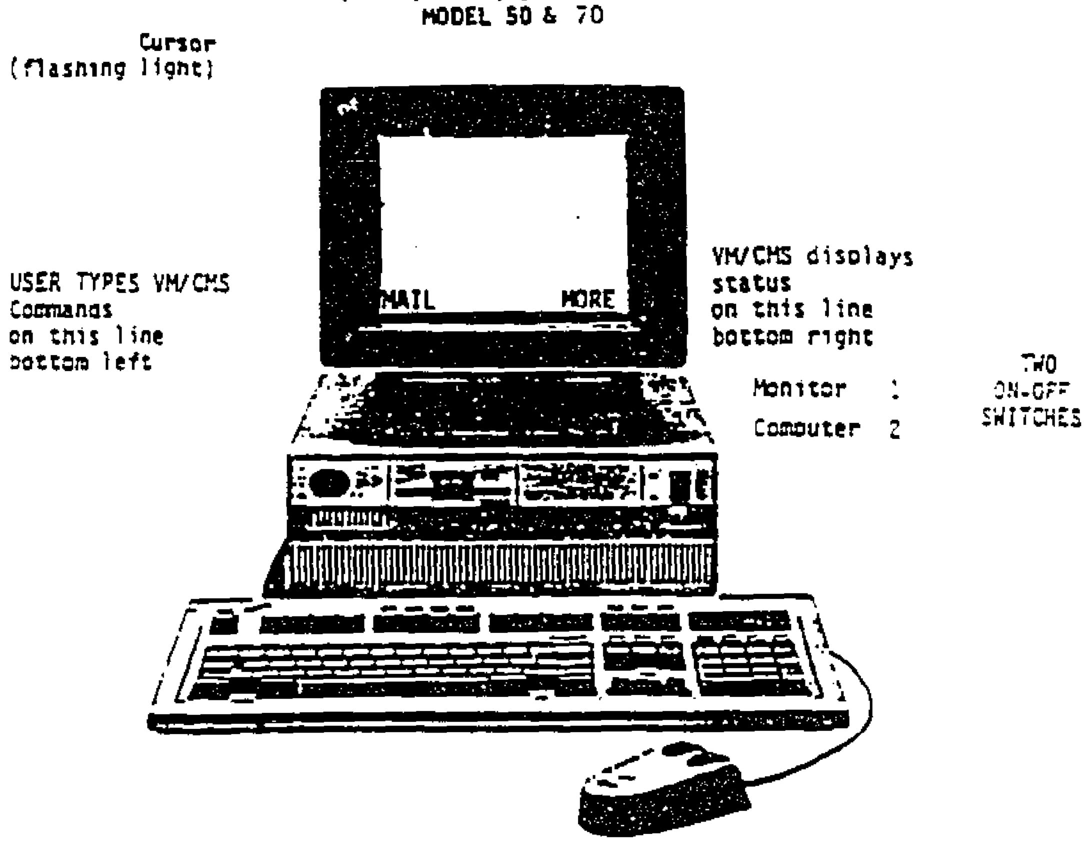
It is recommended that you <u>master the first task</u> before moving on to the second and third tasks.

2.1 LOGGING ON TO THE COMPUTER

The Altoona Campus Computation Center uses the IBM Personal System/2 Model 50Z Video Display Terminal and the Model 70 terminal to communicate with the IBM VM/370 Computers which are located at University Park. The terminals on the second floor of the Computer and Learning Resources Center (CLRC) talk to the U. Park computers via a communications device and a high speed data link. Fundamentally, the terminals on the right side of the second floor are 50Z's and are located in 201 CLRC. The Model 70's are on the left side and are located in 202 CLRC.

Below is a picture of the IBM PS2 terminal with the following features identified: cursor, screen, on-off switches, keyboard, and enter key.

IBM PERSONAL SYSTEM/2
MODEL 50 & 70



<u>IT IS IMPORTANT</u> that you remember a few things when using the PS2 TERMINAL:

After performing a USER ACTION, <u>wait for the SCREEN REACTION</u> before proceeding with the next user action;

The successful user types accurately and verifies the commands that are entered before depressing the ENTER key.

2.2 PS2 LOGON SEQUENCE

When you are seated in front of the terminal, the on-off switches should be off, the screen completely black.

TURN MONITOR 1 SWITCH ON.
TURN COMPUTER 2 SWITCH ON.

Wait for several seconds until CLRC main menu appears. (See p. 17 for main menu)

If main menu does not appear and the screen prompts: Please make a selection: (see p. 18 for alternate connection sequence)

USER ACTION	SCREEN REACTION
depress H key	For Host Connection Menu
....	Host Connection Menu
depress V key	For VM Wait for screen reaction
....	CAC F-Data Switch: FVM
....	Terminal Type:
....	VM/370 ONLINE
....	PSUVM LOGON SCREEN
....	LOGON==
type uid and depress ENTER key (uid-user id from "join" card)	USER ID== ABY101
type password and depress ENTER key (password=password from "join" card)	PASSWORD== (Note: for security reasons, your password will not appear on the screen)
	LOGON AT 14:30:20 EST MONDAY VM/SP CMS etc. A (191) R/O Ready; T=0.12/0.16 14:30:56

As soon as you see the Ready; for Ready prompt, you are successfully LOGGED ON. To disconnect communications or LOGOFF, simply

type LOGOFF and depress ENTER key.

2.2 PS2 LOGON SEQUENCE CONTINUED FROM PAGE 3

*When users log on after PASSWORD expiration, (either as newly activated userid which must change PASSWORD at first use, or at six months intervals) the following two messages will be displayed after the Ready prompt:

VMXSYS011I Your logon password has expired.
VMXSYS372R Select and enter a new password for your userid:

After a new password has been selected, the logon proceeds normally. A new password must be at least naracters, at least 1 alphabetic and 1 numeric, and significantly ___ferent from previous passwords. For example: ABC123.

**If your communications is interrupted, depress CTRL key, ALT key, and DEL KEY and re-do the LOGON SEQUENCE.
Then type BEGIN and depress ENTER key, followed by the + key. (see below)

**
***If the above did not work, redo the LOGON SEQUENCE.
Then type IPL CMS and depress ENTER key.

If at anytime you happen to feel like the terminal is not responding and you notice at the bottom right corner of your screen
MORE...PSUVM or
HOLDING...PSUVM
depress + key (the grey key on right hand side of keyboard)

VERY IMPORTANT INFORMATION DIRECTLY ABOVE

Why?

Because the commands that you enter will not work unless the MORE or HOLDING status is removed.

2.3 LOGON SEQUENCE WITH HDS (Skip this page if you're using PS2)

```
U S E R   A C T I O N                    S C R E E N   R E A C T I O N
---------------------                    -----------------------------

If power light not on, toggle on/off switch on back of crt
depress SHIFT RESET/BREAK key
depress SHIFT F9
depress SHIFT F10
depress RETURN key .....................Data switch
........................................(Please make a selection)
type FVM and depress RETURN key                 FVM
........................................YALE ASCII etc.
........................................Enter Terminal Type:
type HDS and depress RETURN key                          HDS
........................................VM/370 ONLINE

........................................PSU
........................................VM

......................................................RUNNING PSUVM
depress RETURN key                       (screen is all black except
........................................for bottom right) CP READ PSU
type LOGON uid and depress RETURN key
      (uid=user id from "join" card)     LOGON AIP
                                         ENTER PASSWORD
type password and depress RETURN key     (Note:  for security reasons,
  (password=password from "join" card)   your password will not appear
                                         on the screen)

                                         LOGON AT 14:30:20 EST MONDAY
                                         VM/SP CMS etc.
                                         A (191) R/O
                                         Ready; T=0.12/0.16  14:30:56.
```

As soon as you see the Ready; for Ready prompt, you are successfully LOGGED ON. To disconnect communications or LOGOFF, simply

type LOGOFF and depress RETURN key.
If your communications is interrupted, re-do the LOGON SEQUENCE and type BEGIN and depress RETURN key.

If at anytime you happen to feel like the terminal is not responding and you notice at the bottom right corner of your screen

MORE...PSUVM or
HOLDING...PSUVM

depress ENTER key

2.4 LOGON SEQUENCE WITH IBM 3178 Terminal (Skip this page if you're using PS2)

Find terminal marked PSUVM. Depress ENTER and/or toggle ON/OFF key.

USER ACTION	SCREEN REACTION
....	VM/370 ONLINE PSU VM
....	RUNNING PSUVM
depress ENTER key	(screen is all black except for bottom right) CP READ PSUVM
type LOGON uid and depress ENTER key (uid=user id from "join" card)	LOGON AIP ENTER PASSWORD:
type password and depress ENTER key (password=password from "join" card)	(Note: for security reasons, your password will not appear on screen) LOGON AT 14:30:20 EST MONDAY VM/SP CMS etc. A (191) R/O Ready; T=0.12/0.16 14:30:56

As soon as you see the Ready; for Ready prompt, you are successfully LOGGED ON. To disconnect communications or LOGOFF, simply

type LOGOFF and depress ENTER key.
If your communications is interrupted, re-do the LOGON SEQUENCE and type BEGIN and depress ENTER key.

If at anytime you happen to feel like the terminal is not responding and you notice at the bottom right corner of your screen
MORE...PSUVM or
HOLDING...PSUVM
depress CLEAR KEY.

3.0 CREATING, EDITING, VIEWING, AND SAVING A FILE (using a PS2 terminal)
NOTE: If you are using an HDS terminal substitute keypad 6 for F9, ENTER for - , etc.
See page 15 for different keypad layouts.

3.1 CREATING A NEW FILE (with a filename of PROG1)

U S E R A C T I O N	S C R E E N R E A C T I O N
depress ENTER key...................	(Remember, you are actually depressing the ENTER key)
depress ENTER key...................	CMS
type WFOR and depress ENTER key.....	(beginning of file) (end of file)
type *D and depress ENTER key........	NOTE: (Illegal line range) *D will clear all lines, if any, between (beginning and end of file).
type I and depress ENTER key	Now you are in input mode.

(Note: type the next 5 lines, and depress ENTER key after each line.)

```
*THIS IS OUR FIRST FORTRAN EXAMPLE.
*PROGRAM EXAMPLE1 BY TIM LESO.
PRINT, 'THIS IS MY FIRST FORTRAN PROGRAM'
STOP
END
```

depress F9.............................	(end of file) (cursor jumps to bottom left corner ==== "flashing") NOTE: you have just finished input mode and you must SAVE your newly created file.

3.2 SAVING A NEW FILE (with the filename PROG1)

type PUT PROG1 and depress ENTER key..	PROG1- lines transferred=6
type BYE and depress ENTER key	NOTE: Only if you want to quit working now. Then type LOGOFF and depress ENTER key.

4.0 EDITING A SAVED FILE (with the filename PROG1)

```
depress      ENTER key
depress      ENTER key..................CMS
type WFOR and depress    ENTER key
type DIR and depress     ENTER key......PROG1 FILE A1

                                        (bottom right)
                                         MORE...PSUVM
depress + key
type *D and depress    ENTER key
type GET PROG1 and depress    ENTER key..(beginning of file)
                                          (*THIS IS ETC.
                                           PROGRAM.  ETC.;
                                           ETC.
                                          (end of file)
                   NOTE:     Be sure to read 4.6 SAVING AN
                             EDITED FILE, following any editing
                             or your editing work will be lost.
```

4.1 TO CHANGE ANYTHING ON THE SCREEN

```
depress F9.............................cursor jumps to top of
                                       screen from bottom left
                                       corner.
depress up arrow key...................moves cursor up
depress down arrow key.................moves cursor down
depress right arrow key................moves cursor right
depress left arrow key.................moves cursor left

move cursor directly over change to be made and retype it
depress F9
```

NOTE: Other methods to change/insert <u>one</u> line is:

```
depress F9.............................move cursor to line to
                                       be changed in this
                                       example, we will insert
                                       4 spaces before the word
                                       PROGRAM
depress F9
type CH/PROGRAM/    PROGRAM/and depress    ENTER key
or use INS key and DEL key ............to insert/delete one
                                       character at a time.
```

4.2 TO INSERT ONE NEW LINE ON THE SCREEN

depress F9..............................cursor jumps to screen from bottom
move cursor to line directly above the line-to-be-inserted and position cursor at the beginning of that line .....for example
===== * THIS IS ETC.
the cursor should be over the asterisk. it will flash.
depress F5..............................* THIS IS ETC.
(-one blank line-)
PROGRAM ETC.
type new line and depress ENTER key

4.3 TO INSERT MORE THAN ONE NEW LINE ON THE SCREEN

depress F9..............................cursor jumps to screen from bottom move cursor to line directly above the lines-to-be-inserted and position cursor at the beginning of that line
depress F9..............................cursor jumps to bottom left
==== position

type I and depress ENTER key.........(* THIS IS ETC
(-blank lines appear to the bottom of the screen-)

type each new line and depress ENTER key
when you finished inserting the last line
depress F9..............................cursor jumps from line to bottom left

4.4 TO DELETE ONE LINE FROM THE SCREEN

depress F9............................cursor jumps to screen from bottom left position.
position cursor to the line-to-be-deleted with arrow keys
depress F6............................and it disappears!!
depress F9............................cursor jumps to bottom left ==== position

4.5 TO DELETE MORE THAN ONE LINE

depress F9............................cursor jumps to screen from bottom left position.
move cursor to first-line-to-be-deleted
depress F9 ....cursor jumps to bottom left ==== position.
type .,+2 D and depress ENTER key....deletes 3 lines starting with line pointed to by cursor in previous step. (note, if you wanted to delete 6 lines, then use +5, 9 lines, +8, etc.)

(saying the above in plain english-- type PERIOD COMMA PLUS TWO SPACE D)

4.6 SAVING AN EDITED FILE (with the filename PROG1)

<u>immediately</u> <u>after</u> <u>editing</u> a file, you should <u>save</u> <u>the</u> <u>changes</u> or your editing work will be lost.

IMPORTANT

---

type PUT PROG1 and depress ENTER key....

PROG1-lines transferred=10

5.0 TO VIEW LINES IN A SAVED FILE (with the filename of PROG1)

(NOTE: Section 5 assumes you are in the WFOR environment.)

type *D and depress ENTER key
type GET PROG1 and depress ENTER key....
(beginning of file)
etc.

type 9 and depress ENTER key......(the first page is displayed)

depress F2...........................(display forward to 2nd page)

depress F2...........................(display forward to 3rd page)

etc.
depress F1...........................(display backward to 2nd page)

depress F1...........................(display backward to 1st page)

etc.
type $ and depress ENTER key ......(display last page of file)

(end of file)

type BYE and depress ENTER key.....your done for today!!!!!! so logoff.

type LOGOFF and depress ENTER key..the system will bid you farewell with time, etc.

6.0 RUNNING YOUR PROGRAM AND MAKING A LISTING

(NOTE: Section 6 assumes you are in the CMS environment.)

6.1 TO RUN A SAVED PROGRAM (with the filename PROG1)

```
depress    ENTER key
depress    ENTER key........................CMS
type WFOR and depress    ENTER key
type DIR and depress    ENTER key...........PROG1 FILE A1
                                                   MORE...PSUVM
depress -

type *D and depress    ENTER key

type GET PROG1 and depress    ENTER key.....
                                            (beginning of file)
                                             etc.
type RUN and depress    ENTER key...........
                                            EXECUTION BEGINS...
                                            THIS IS MY FIRST
                                            FORTRAN PROGRAM
                                         ...EXECUTION ENDS
                                                 MORE...PSUVM
depress + ..................................
                                            (beginning of file)
                                             etc.
type BYE and depress    ENTER key
```

5.2 TO MAKE A LISTING OF A SAVED PROGRAM (with a filename of PROG1)

```
depress    ENTER key
depress    ENTER key........................CMS
type WFOR and depress    ENTER key
type DIR and depress    ENTER key............PROG1 FILE A1
                                                  MORE...PSUVM
                                             NOTE:  Remember

                                             PROG1 FILE A1
                                             for later.

depress ÷

type BY: nd depress    ENTER key...........
                                             Ready; T=0.13/0.30
                                             etc.
type LISTOFF PROG1 FILE A1 and depress    ENTER key
depress    ENTER key                         Selects Default
                                             printer device
depress    ENTER key                         Selects Default
                                             site to send output


                                             listing will be
                                             printed on the
                                             printer behind the
                                             counter of the
                                             CLRC.  Ask
                                             Attendant for Your
                                             Listing, or check
                                             storage bins.
```

6.3 TO MAKE A LISTING OF OUTPUT GENERATED DURING A PROGRAM'S RUN (with the filename PROG1)

```
depress     ENTER key
depress     ENTER key.........................CMS

type LISTOFF CONSOLE and depress     ENTER key     CONSOLE
                                                   SPOOLING ...
                                                   STARTED
     depress    ENTER again        to choose default
     depress    ENTER again        print options
type TYPE PROG1 FILE A1 and depress
    ENTER key................................program lists on
                                             screen

type WFOR and depress     ENTER key
type DIR and depress    ENTER key.............PROG1  FILE  A1

                                                MORE...PSUVM
depress +

type *D and depress    ENTER key
type GET PROG1 and depress    ENTER key
type RUN and depress    ENTER key
                          .............EXECUTING...
                                             etc.
                                       ...STOP
                                                MORE...PSUVM

depress +

type BYE and depress     ENTER key..........(you must logoff to
                                             stop listing, so do
                                             so!!!)
type LOGOFF and depress     ENTER key.......CONNECT=00:33:22
                                             VIRTCPU=LOGOFF AT
                                             time of day.


                                             (your program's
                                             execution that
                                             appeared on the
                                             screen will be
                                             written to the
                                             printer in the
                                             CLRC.)
```

(See Sample Listing on Page 21)

So go to the Counter and ask the work study student for your listings, or check in the storage bins.
That is all. Be sure terminal is the way you found it.
**Screen all black.
Remember to turn off monitor switch, but leave computer switch on.
Thank you.

7.0 References

The Pennsylvania State University Computation Center, PSU Guide to VM/CMS, Parts 1 and 2, August 1990.

Dirksen, P. H. and Welsch, J. W. WATCOM FORTRAN: Tutorial and Reference Manual, Waterloo, Ontario, Canada: WATCOM Publications, 1983, 196 + xii pp., ISBN 0-919884-24-5.

Boswell, F. D., Grove, T. R., and Welch, J. W. WATCOM PASCAL: Tutorial and Reference Manual, Waterloo, Ontario, Canada: WATCOM Publications, 1983, 147 + viii pp., ISBN 0-919884-26-1.

Boswell, F. D., et al. WATERLOO MICROEDITOR, WATSOFTPRODUCTS, INC., A subsidiary of the University of Waterloo, 1982, 38 + iv pp., Waterloo Computing Systems Newsletter.

7.1 HDS, PS2, and IBM 3178 PROGRAMMED FUNCTION KEY (PF) LAYOUTS AND KEY EQUIVALENCES

HDS			IBM PS2	IBM 3178		
F13	F14	F15	F13 F14 ... F24(Shift)	1	2	3
7	8	9	F1 F2 ... F12	4	5	6
4	5	6		7	8	9
1	2	3		10		

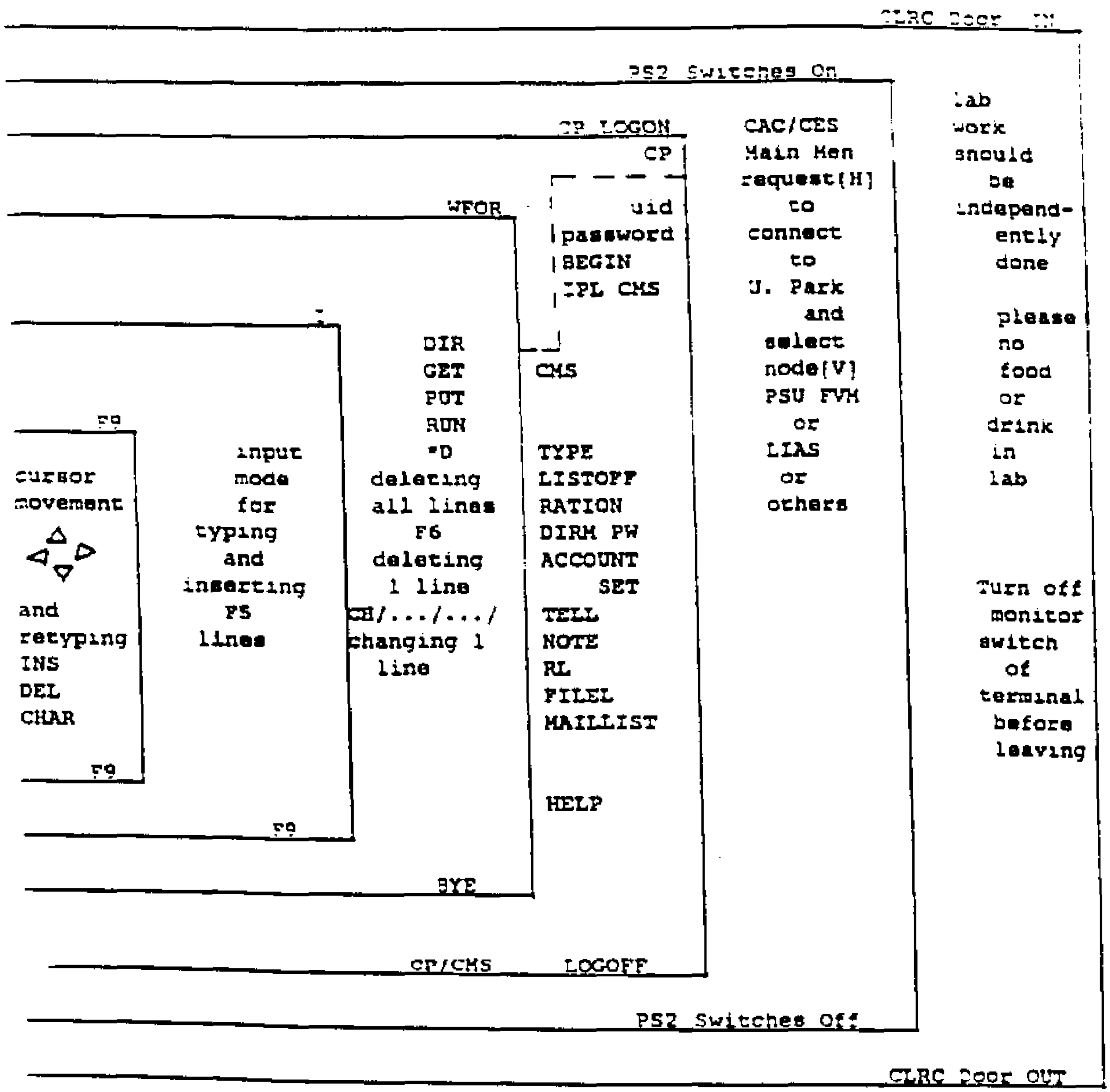
E.G., HDS Keypad 6 = PS2 F9 = IBM Keypad 9

HDS RETURN key = PS2 ENTER KEY = IBM ENTER key

HDS ENTER key = PS2 + Key = IBM CLEAR key

(See Page 2 for PS2 keyboard layout.)

8.0 The Many Environments of CompSci IBM LAB
(SCREEN, INPUT, FORTRAN, CMS, PS2, CLRC)



LEGEND:

I ===> Getting into Input Environment

F9 ===> Getting Out of Input Environment

or

F9 ==> Command Line to Screen

F9 ==> Scr: to Command Line

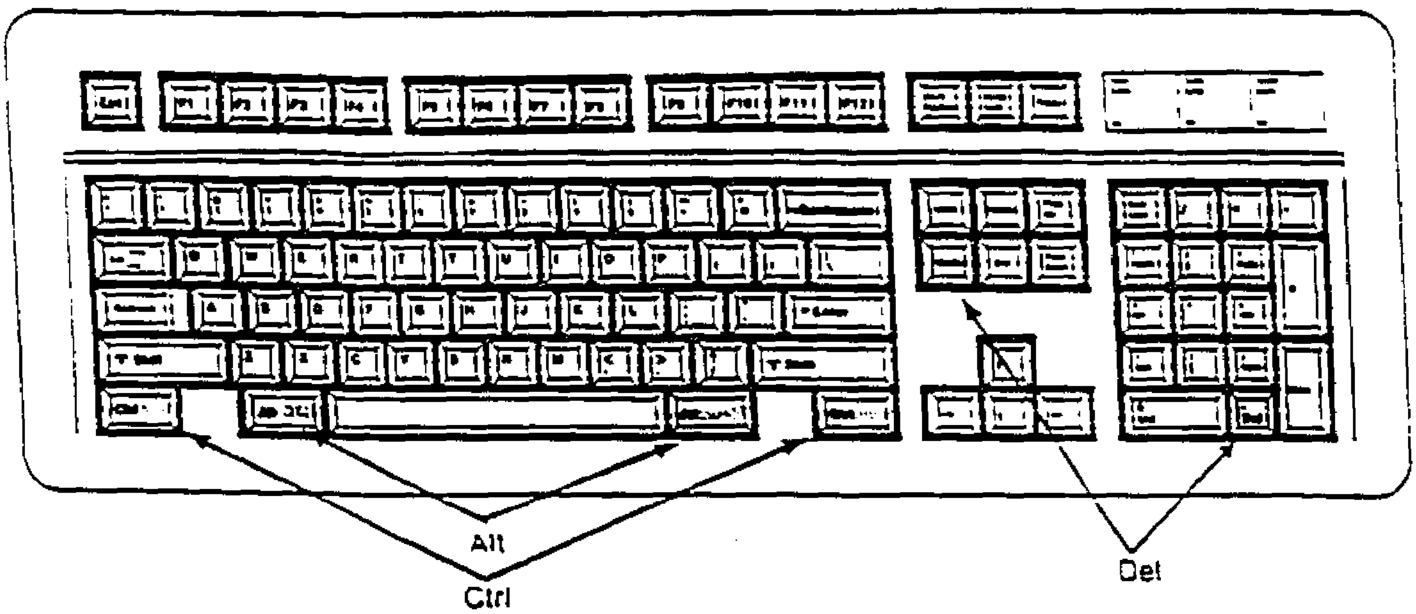
3.1 CAC/CES 502 CONNECTION TO PSUVM

PSU Main Menu

(A) A: Drive
(H) Host Connection Menu
(I) Information Menu
(L) Language Menu
(N) Network
(U) Utility Menu
(W) Word Processor Menu - Word Perfect, Microsoft Word
(X) Exit

Host Connection Menu

(F) File Transfer Program
(K) VM Keyboard
(L) Lias Connection
(V) VM Connection
(X) Exit - Return to Main Menu



ALTERNATE CONNECTION TO PSUVM

Press (Alt/Prtscrn) to return to PSU Altoona Menu System...

Connecting to VM. Wait for VM logo...

PSUNET Altoona
Node 20 Port 78

Please make a selection : FVM
Connected..

CAC S-DataSwitch
Request:FVM

Connected

enter terminal type: IBMPC

9.0 E-MAIL OVERVIEW

* DEMONSTRATION BY LESO IN 129 EICHE

* PRACTICE SENDING AND RECEIVING E-MAIL TO YOUR PARTNER IN 201 or 202 CLRC

DO 1 -> 7 BELOW:

1. LOGON TO PSUVM WITH USERID
   TYPE PASSWORD * (* Depress NL ENTER Key hereafter referred by an *)

2. SEND ON-LINE MESSAGE (2 LINES) TO PARTNER
   TELL PARTNER'S USERID SHORT MESSAGE *

3. COMPOSE AND SEND LETTER (MEMO) TO PARTNER
   NOTE PARTNER'S USERID *
   TYPE SUBJECT OF MEMO *
   TYPE L... S OF MEMO *
   ON A BLANK LINE *
   DEPRESS F5 ==> SEND

4. READ LETTER FROM PARTNER (VIEW YOUR READERLIST)
   RL *
   DEPRESS F11 ==> PEEK

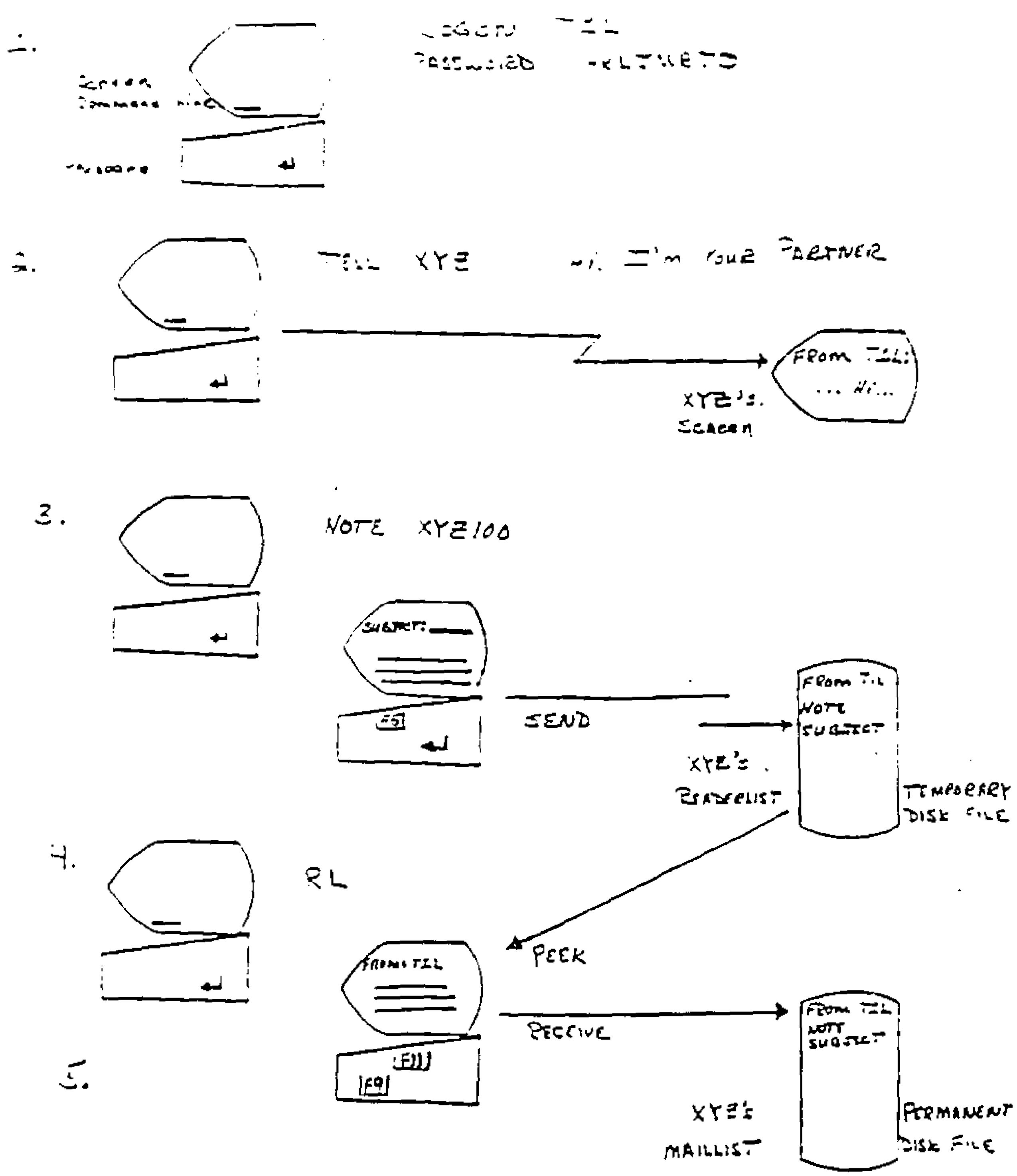
5. SAVE ...TER FROM PARTNER (RECEIVE IT AND FILE IN MAILLIST)
   DEPRE... F9 ==> RECEIVE

OR

6. DISCARD LETTER FROM PARTNER (THROW IT AWAY)
   TYPE DISCARD *

7. LOOK AT OLD LETTERS
   MAILLIST *
   DEPRESS F3 ==> TO QUIT

IMPORTANT COMMANDS AND KEYS



[illegible] SAMPLE LISTING [illegible] OUTPUT GENERATED DURING RUN [illegible]

```
CONSOLE SPOOLING TO NETWORK STARTED
Ready; T=0.04/1.00 13:40:43
type pg1 file a

* This is our first fortran example
* Program example1 by Tim Leso
print*, 'this is my first_fortran program'
stop
end

Ready; T=0.00/0.01 13:42.05
wfor
ACLDD      JOB       A1
ALL        NOTEBOOK  A0
CHAPMAN    REPLY     A1
COBF       JOB       A1
COBF       OUTPUT    A1
COBFOOD    OUTPUT    A1
C2         JOB       A1
C2         OUTPUT    A1
EBDASH6    FILE      A1
GRADE3     FILE      A1
LAB418     FILE      A1
LASTING    GLOBALV   A1
LESOT      JOB       A1
LESS       JOB       A1
LESS       OUTPUT    A1
LP         HEADER    A1
LPRINT     EXEC      A1
MACROXX    ASSEMBLE  A1
MYFILE     FILE      A1
OHMS       FILE      A1
PROB705    FILE      A1
PROCPROJ   FILE      A1
PROFILE    EXEC      A1
PROGPROC   FILE      A1
PROG7      FILE      A1
PROJECT2   FILE      A1
SEARCH     FILE      A1
SPACHT     REPLY     A1
STTEST     FILE      A1
TFILE      FILE      A1
TIMS       JOB       A1
TIMT       JOB       A1
TIMT       OUTPUT    A1
TXS110     FILE      A1
T1L        NETLOG    A0
VANCAMP    REPLY     A1
7895       PEEK      A0
PG1        FILE      A1
Executing
 this is my first fortran program
  ..Stop
Ready; T=0.03/0.08 13.42.46
logoff
CONNECT= 00:06:29 VIRTCPU= 000:01.13 TOTCPU= 000.01.44
LOGOFF AT 13:42:52 EDT [illegible]DAY 08/28/90
```

VITA

Leso, T. J. (Tim)	Instructor of Computer Science and Engineering, Penn State University, Altoona Campus.

EXPERIENCE

1980-Present:	Instructor of Computer Science for Baccalaureate and Associate Degree Programs, Eberly College of Science.
1978-1980:	HRB-Singer, Inc., State College, PA. Principal Engineer.
1975-1978:	The Mitre Corporation, McLean, VA., Member of Technical Staff.
1972-1975:	TRW, Inc., McLean, VA., Member of Technical Staff.
1973-1977:	The American University, Washington, D.C. Part-time Lecturer, Department of Mathematics, Computer Science and Statistics.
1962-1966:	U.S. Air Force, Great Falls, MT. E4, Ballistic Missile Analyst Technician, Minuteman Missile System.

EDUCATION

1994:	Penn State University, PhD, Instructional Systems.
1971:	Penn State University, MS, Computer Science.
1970:	Penn State University, BS, Computer Science.
1970:	Penn State University, BS, Mathematics.

PROFESSIONAL

Special Security Clearance Inactive.
Member ACM, CMPSC Educators, Chamber of Commerce, Southern Alleghenies Small Business Computing, Academic Affairs Committee, Honors Committee, Technical Advisory Committee, Center for Academic Computing Faculty Advisory Committee.
Fourteen Technical Publications Authored/Co-authored.
Ph.D. Thesis - Presentation Strategies, Aptitude, and Computer Anxiety.
Master's Paper - Dynamic Storage Allocation System.
Coordinator of Kanesatake's Summer Computer Camp for Children.

PERSONAL

Mr. T. J. Leso, Spouse, Jeanne Spicer, Son, Mickey.
P.O. Box 235
Warriors Mark, PA 16877
(814) 632-3155