A Study of the Effects on Attitude and Achievement of Mode of Processing in a Secondary School Course in Computer Programming

by

William Joseph Moulds

Dissertation submitted to the Faculty of the Graduate School
of the University of Maryland in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
1976

APPROVAL SHEET

Title of Thesis: A Study of the Effects on Attitude and Achievement of Mode of Processing in a Secondary School Course in Computer Programming

Name of Candidate: William Joseph Moulds
Doctor of Philosophy, 1976

Thesis and Abstract Approved: Henry H. Walbesser

Henry H. Walbesser
Professor of Education

Date Approved: August 4, 1976

Abstract

Title of Thesis: A Study of the Effects on Attitude and Achievement of Mode of Processing in a Secondary School Course in Computer Programming

William Joseph Moulds, Doctor of Philosophy, 1976

Thesis directed by: Dr. Henry Walbesser
Professor of Education

This study was designed to compare the effects on attitude and achievement of frequency of association with the computer and turn-around time in a beginning course in computer programming in the secondary school. The treatments frequency (F) and turn-around time (T) each existed at three levels. The frequency treatment involved the number of hours per semester the students were in direct association with the computer. The three levels were: H0 - one hour per semester; H1 - one hour per week per semester; and, H3 - three hours per week per semester. The second treatment, turn-around time, dealt with the time delay experienced by the student between submittal of his program deck and return of his printout. This treatment existed at the following three levels: PI - printouts returned on the same day as submittal of the program; PD - printouts returned in about 24 hours; and, PW - printouts returned in about one week.

At the beginning of the course, each student was administered the Aptitude Test for Programming Personnel to determine his aptitude in programming and to serve as a covariate in the analysis of covariance. A multiple choice final examination was administered to all students at the end of the 16 week semester to test their understanding of

FORTRAN programming concepts. A 3 x 3 analysis of covariance design, using the aptitude scores for covariate, was used to analyze the results. This analysis served as the basis for conclusions to the following research hypotheses.

1. More frequent access to the computer increases performance.
2. Immediacy of feedback increases student achievement.

The first hypothesis is supported by the findings of this experiment. Students in direct association with the computer most frequently (three hours per week) scored significantly higher on the final examination at the .05 level than students in either of the other two levels of this treatment.

The second hypothesis is also supported by the findings. Students who received their printouts in less than one day scored significantly higher at the .05 level on the final examination than students receiving their printouts in about one day. Those receiving their printouts in about one week scored significantly lower than either of the other two groups.

The effects on attitude were tested using a Likert-type scaled instrument. Analysis of covariance, using the aptitude score as covariate on a 3 x 3 design, was used to analyze the results which served as the basis for conclusions concerning the following research hypotheses.

3. More frequent access to the computer results in a more positive attitude by the students toward the computer.
4. Immediacy of feedback of computer programs results in a more positive attitude by the students toward the computer.

The third hypothesis is not supported by the findings of this experiment. Students who were in the direct association with the computer three hours per week scored significantly lower at the .05 level on the attitude instrument than students who were in the one hour per semester group.

The fourth hypothesis is not supported by the findings of this experiment.

These conclusions are discussed in terms of earlier research in the effects of mode of processing on student learning of programming concepts and feedback schedules. In addition, suggestions for further research are offered.

ACKNOWLEDGEMENTS

I wish to express my gratitude to Dr. Henry Walbesser for his patience and concern, and for sharing of his ideas and knowledge in the area of secondary mathematics education. A special thank you is extended to Dr. John Giblette, for his technical assistance with the design and evaluation of the results of the experiment. Dr. James Fey and Dr. Harry Tunis devoted much time and shared many ideas with me in developing the study, as did Dr. Robert Hardy. For additional help with the evaluation of results I am indebted to Dr. C. M. Dayton.

The principal of Baltimore Polytechnic Institute, Mr. William Gerardi, very graciously granted permission for this study to be done at his school. I am indebted to the following members of the Mathematics Department: Billie Eskut, James Newton, M. Carroll Paugh, Richard Whippo and Irvin Yaffe.

A very special vote of thanks is extended to Mr. Jacob Schuchman. To my wife, Lee, and children, Bill and Cara, my appreciation of their love, patience, and faith in me, as well as their help, can never be fully expressed.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

A survey conducted in 1970 by the American Institute for Research under the auspices of the National Science Foundation showed 12.9 per cent of public high schools reporting were using computers in their instructional programs (Atchison, 1973). More recent surveys show that more than 20 per cent of United States public high schools are using computers. In addition, an ever growing number of junior high schools and elementary schools are using computers (Atchison, 1973).

The Pittsburgh public school system recently began an experimental program involving high school students using computer programming to aid in learning mathematics (Dwyer, 1975). The experiment was conceived in response to the recent wave of criticism directed at the "New Math" and its "needless obfuscations". Dwyer reports, referring to the student subjects of the experiment

> They've learned that writing, debugging, and running programs is an excellent tool for weeding out the irrevelant and trivial. It's almost always true that a student can improve upon someone else's idea, given the chance to work with that idea on a computer.

The computer has become an important asset on the American college campus. It is indispensable to administrators and researchers, and its role as an educational device is still expanding. The importance of the computer in education was recognized by Dr. John Kemeny, (1970), when he stated

> Learning to use a high speed computer should be an essential part of a liberal education.

Long before Kemeny's perceptive statement the Committee on the Undergraduate Program in Mathematics published a report in The Mathematics Teacher, (1965), which stated

> Since modern automata such as computers are playing an increasingly important role in everyday life and in education, it is important that, early in his career, the undergraduate achieve some intellectual understanding of these devices and of methods of using them.

The President's Science Advisory Committee prepared a report on the role of the computer in higher education. The committee reported "Computing increases the quality and scope of education".

Weeg (1970) stated this same concept in different terms.

> The more exciting motivation of working with real data on real problems, with instant reward (or lack of it) for success, and with the student's assuming full responsibility for his success or failure, all contribute to greater teaching.

It is evident that the computer has come to be recognized as an effective instructional device at every level of education. With such widespread use of computers for instructional purposes, it seems appropriate to assume the worthwhileness of computer programming as a course in the high school curriculum, and to investigate the most effective methods of presenting such a course.

In 1969 the Baltimore City Public School (BCPS) system recognized the need for including in the curriculum a course in computer programming. With the cooperation of IBM, a mobile computing unit was stationed at the Baltimore Polytechnic Institute, a public high school within the BCPS system, for one month as an experiment to determine if sufficient interest existed among high school age students. During this month, all mathematics students at the school (BPI) were introduced to the system via a talk and demonstration, conducted by a team of IBM employees. In addition, several interested students actually learned how to program the computer to solve problems in mathematics and science classes. Based on the one month experimental results, the BCPS decided to rent and install a computing system for the purpose of teaching computer programming to high school students throughout the city system. As a result, the IBM 1130 computing system was installed at BPI in the Fall of 1969, and the first classes were begun immediately thereafter.

The course has become a popular one with students throughout the system. Each year, students from several public secondary schools throughout the BCPS system use the IBM computer installation at BPI. For the most part, they are students enrolled in an introductory course in FORTRAN IV programming language. In the current school year, there are nine high schools using the computer, including BPI, and approximately 450 students.

Several concerns have arisen as a result of the physical arrangement of the schools in relation to the computer facility. Among these concerns are the following:

1. Is a single large computer located at one site preferable to several smaller and therefore less expensive computers located at each of the schools using the present facility?
2. What is the most effective method of managing student use of the computer and processing of computer programs?

Currently there are three modes of access to a computer in widespread use: time-sharing, on-site batch, and remote batch. Gumm (1969) conducted a survey of existing educational facilities. He found four practical plans for managing the processing of computer programs, using the three modes named above.

Plan I: Courier service of prepared computer programs to a computer center for processing. Under this method of operation, the student prepares the program deck of computer cards at the school in which he is enrolled. Presently, there are several ways of preparing such program decks. The student may use a keypunch machine to punch the cards, or have the cards punched from a written program. Alternatively, the student may use a type of programming instruction card called a "mark sense" card, or "optical mark" card. Using these cards, the student would mark with a soft, dark pencil on the card the program instructions, using specially coded characters. Such cards can be read and interpreted by specially equipped computers. When decks of cards are prepared by whichever means described above and sent to the computer for processing in bulk this is referred to as "batch processing".

Plan 2: Transporting students to a computer facility, or computer center. School districts employing this system usually are able to transport the students at regular but infrequent intervals. The students receive most of their programming instruction at the school in which they are enrolled. When arriving at the computer center they spend their time processing and correcting programs.

Plan 3: Remote terminals located in several schools, hooked up to a single, centrally-located computer. The terminals are teletypewriter machines, wired to the computer by telephone lines. Students communicate their programs, one student at a time, typing one line at a time on the teletypewriter. After each line of typing there is a pause while the computer assimilates the line of instruction, then signals the student to type the next instruction. This mode of student-computer communication is referred to as "on-line" or "time-sharing" processing.

Plan 4: Acquisition of a computer of relatively small size and therefore inexpensive, for use "on-site". This plan allows the student to see his program being processed and may even permit him to operate the computer (referred to as "hands-on").

For the most part, educational systems use a combination of batch processing on-site and remote time-sharing processing. This study is concerned with processing computer programs in a batch mode rather than time-sharing. There are two reasons for the selection of batch processing mode over time-sharing. First, in a study comparing time-sharing and batch processing modes of processing, Pack (1970) found no significant difference in ability to learn programming between the two modes. Second, most schools using a computer already own, lease, or otherwise have access to a computer which uses the batch-processing mode. Rudolph (1972) found among the five per cent of Illinois schools responding to her survey that only twelve used terminals and the time sharing mode. The BCPS system uses the batch processing mode with the computer center at BPI.

Statement of the Problem

This study investigated the effects on student attitude and achievement in a course in computer programming presented under various modes of batch processing. Two questions are posed in the study:

1. Does having direct access to a computer have any effect on student attitude and achievement in a course in computer programming?
2. Is there any differential effect on student attitude and achievement in a course in computer programming caused by the time lapse between submittal of a program for processing and return of the printout?

The students chosen to participate in this study are enrolled in Baltimore City Public Schools, in a course in computer programming. Currently, there are seven high schools using courier service to have student programs processed at the computer center in the batch mode. The computer center is located at Baltimore Polytechnic Institute, and consists of an IBM 1130 computing system. The center school, BPI, also offers courses in computer programming to its students. Once each semester the non-center schools transport their students to the center to see the facility. For the most part, however, the students seldom see the computer or the processing of their programs. The time lapse between sending off the programs and receiving printouts from the computer center varies greatly. In some cases, the teacher carries the batch of cards to the computer center after school, returning them to the students the following day. Thus, the time lapse, or "turn-around time", is usually less than 24 hours. Other schools must rely on a system-wide courier service which may cause time delays of up to five days. The teachers involved are presently attempting to institute express courier service which would operate between only the schools using the computer. It is hoped that such service, if instituted, will shorten turn-around time, thereby enhancing learning and interest for the students.

The second chapter of this paper reviews the literature and research related to this study. The basic concerns of this study deal with two aspects of computer program processing for students: (1) accessibility to the computer; and, (2) time delay between submittal of program decks and return of printouts (turn-around time). Thus, the literature reviewed surveys the studies related to these aspects of computer programming, beginning with the more recent studies first. Other studies, related to the computer as an educational device, are also reviewed.

CHAPTER II

REVIEW OF RELATED LITERATURE

Mode of Processing

There is a wealth of literature related to the computer, most of which describes studies concerning computer-assisted instruction. Surprisingly, however, there is little material related to processing of computer programs and the effect of processing on student achievement and attitude. Many of the studies dealing with processing of programs are concerned with comparisons between batch or time-sharing systems. Some of these are discussed in this review, inasmuch as they were somewhat related in terms of achievement and attitude.

In one of the studies comparing batch and time-sharing modes of processing, Sackman (1968) stated

> Amid all these portents of the dominating role that computer programming will play in the emerging computer scene, one would expect that computer programming would be the object of intensive applied scientific study. This is not the case ... a widening and critical lag threatens the industry and the profession.

The report went on to compare batch processing and time-sharing modes from an economic point of view and in terms of its relation to the body of knowledge of learning theory. Table 1 represents the comparison of the two systems financially.

The determination of which system has an economic advantage over the other would depend upon several factors, including the purpose for which one intends the system, the capacity needed, the suitability of the system in terms of adaptation to existing physical site, and so on.

TABLE 1

Estimated Annual Budget by
Type of Access

Type	Capacity*	Languages	Cost/Year
Time-sharing (1 - 4 users)	4k - 12k	BASIC	$16250
Time-sharing (4 - 8 users)	12k - 16k	FORTRAN BASIC	$38310
Batch	16k+	FORTRAN BASIC APL COBOL	$39233

*Capacity refers to memory and processing capability; 8k and under is usually considered relatively small capacity.

The data would appear to make a case for batch mode processing for users anticipating more than 4 users at one time, based on cost effectiveness. Actually, most installations presently in use employ both time-sharing and batch processing modes.

A study comparing two methods of utilizing the keypunch and the computer was conducted in Baltimore County Public School system, (Verardi, 1971). The experimental class had its keypunching of programs done by the Data Processing Department of the school system. The control class did their own keypunching of programs. The objective of the course was to use the computer as a tool in learning mathematics. The control class was bused once a week to the computer center where their programs were executed for them by employees of the system. On the other hand, the experimental class saw the computer facility just once, at the beginning of the study. Both groups received an achievement test at the end of the school year to determine how well they learned to write programs. The results of the test showed that although the control class mean scores were higher than the experimental class mean scores, they were not significantly higher at the .05 level. Verardi concluded that the results imply no difference in programming achievement between the control and experimental groups.

Katz compared students in secondary schools enrolled in Algebra II and computer programming to those enrolled in Algebra II without computer programming (Katz, 1971). The students taking the Algebra-Computer course were divided into two groups: those who wrote and processed programs and those who only wrote the programs, without processing. Katz compared the three groups on an achievement test. The results showed no significant differences between the groups. Katz did, however, report two significant findings. First, those students who wrote and processed their own computer programs scored significantly lower on the Cooperative Mathematics, Algebra II, test. Katz offered no explanation of this apparent phenomenon. Second, students who wrote their own programs, without processing, did best of all.

Another study which investigated the effects of various methods of program processing was done by Pack (1970). He studied the efficiency with which college bound students learn a problem-solving language under three modes of program processing: time-sharing, using a teletypewriter; "quick-batch" (8 minutes average time lapse); and, finally, "slow batch", (24+ hours average time lapse). Pack found no significant difference in the student's ability to learn the programming language among the three modes. However, the students' responses to a survey seemed to indicate a preference for time-sharing using a teletypewriter.

Cocker (1970) reported on an arrangement with a vocational and technical school under which an IBM 360 was used to process students' programs. The programs were sent to the computer site for processing whenever the programs could be fitted into the computer schedule. The result was one day "turn-around time", meaning the processed programs were returned in 24 hours. The students did their own keypunching at their own school. Gocker reached the conclusion that this joint venture was "successful", though no supportive data was included in the report.

A two-year study was conducted by Hrasky (1969), exploring the use of time-sharing and other computer program processing systems. After the two year study was completed, he concluded that a computer housed in the students' own school was superior to any other method of processing programs. This accessibility of the computer, according to Hrasky, permitted the students to gain valuable "hands-on" experience. Again, as in the Gocker study, no supportive data was supplied to substantiate the conclusion.

The Latin School in Chicago obtained a grant to install and utilize a computing system for instructional purposes. The computer was used by any and all interested students. Two problems were reported by the school. First, only a single keypunch machine was rented, thereby creating problems for students desiring to punch their programs. Second, the faculty was inadequately trained prior to the installation of the computer (Bucholz, 1969).

An opinion was offered by Hanson (1968) to the effect that teachers and administrators of data processing were not in complete agreement concerning the necessity of having the equipment on hand in order to promote an understanding of basic automated business data processing concepts. He reported that, according to some educators, the availability of equipment develops and maintains interest, motivation, and a desire to attain maximum achievement. Others, however, were of the opinion that the same ultimate objectives could be achieved through other vicarious experiences. Hanson conducted a two-part study to ascertain the significance of machine time on learning selected data processing concepts and on student attitudes. The two parts consisted of the teaching of unit record concepts, and of the teaching of computer concepts. In both parts, those students having "hands-on" experience scored higher on an achievement test than those not having "hands-on" experience. However, the differences were not significant. The group having hands-on experience had a significantly more positive attitude toward course expectation fulfillment. They also rated methods of instruction significantly higher.

In a report appearing in National Business Education Quarterly, (1967), Miami-Dade Junior College indicated that it had changed its hands-on approach, replacing it with a batch processing system. Under this new approach, the student could elect to use a teletypewriter terminal if one so desired. The administrators involved offered the opinion that this new approach was not only the most efficient economically, but was the only way a small school with limited funds could afford to conduct an instructional program in computer programming. They felt that the design of modern computing systems with multiprogramming capabilities makes the computer center concept the most viable arrangement.

Reno, Reno, and Saul, (1967), reported on a project conducted by the Euclid, Ohio, Board of Education. In this study, about seventy students used a computer housed at Case Institute of Technology in Cleveland, Ohio. At first, the students were bused to the computer facility to process their programs. This was soon discontinued, due to the expense involved in terms of both money and time. Other reasons were given, such as scheduling problems, disruption of other courses, and imposition on the computer center. The research project continued employing a different method for processing student programs. The programs were delivered to the computer center via courier service. This method was considered slow and clearly showed, in the opinion of the researchers, the disadvantage of not having a computer on site. The school board investigated the possibility of leasing or purchasing a computer for on-site installation, but abandoned the project as too expensive.

A report by Hoffman, et al, (1965), presented the various possible alternatives available for instituting a programming course. If a batch process method was selected, but the computer was not housed in the school, they suggested using a courier service or transporting the students to the facility. The best way to begin a programming course, according to the Hoffman study was by use of the courier service method of processing programs. However, they asserted that the service had to be expeditious and simple. The advantage of such a system, according to the report, was the relatively low cost. The disadvantage was the delay for the student in receiving processed programs. Transporting students to the computer center was also discussed in the report. This system, it was felt, had the advantage of the motivating effect of hands-on experience, but the cost was prohibitive. No mention was made in the article about keypunching of programs.

Computation Planning, Inc., prepared a report for the U. S. Office of Education, comparing time-sharing and batch processing systems. According to this report, although many people believed the pedagogical effectiveness was greater in time-sharing, no evidence existed to support such a hypothesis. The report presented arguments, without supportive evidence, in favor of batch processing when the batch usage was coupled with quick turn-around time.

Summary

Very few of the articles and research projects reviewed presented substantive evidence in favor of one mode of processing over another. However, at least one study indicated that two modes of processing, time-sharing and transportation of pupils to a computer center, had very definite disadvantages. Experience has shown that transporting of students to a computing center has proved costly and unwieldy, (Hrasky, 1969), Miami-Dade Junior College, (Reno et al, 1967), (Hoffman, et al, 1965) and the report of Computation Planning, Inc. Although some students expressed a preference for time-sharing over batch processing mode, (Pack, 1970), other reporters stated that problems were encountered using this mode, (Verardi, 1971), (Hrasky, 1969), Miami-Dade Junior College, and the report of Computation Planning, Inc. The greatest problem encountered under time-sharing mode was the fact that only one student at a time could process a program unless several teletype terminals were used. This in turn increased the cost per pupil substantially.

Batch processing with relatively short turn-around time seemed to receive favorable reports in terms of operating efficiency and cost, as reported by Hoffman et al (1965), Computation Planning Inc., Miami-Dade Junior College and Verardi, (1971). Again, however, disadvantages for this system do exist, primarily in terms of student inability to gain hands-on experience due to physical limitations. Such hands-on experience is vital to enhance student motivation and interest, as reported by Pack, (1970), Hrasky, (1969), and Hanson, (1968).

The general consensus seems to be that an on-site computer facility which is large enough to handle FORTRAN IV programming language, and is also relatively inexpensive, is the most desirable situation. Finding such a computer facility is difficult at best. Thus, most educational systems have elected to purchase, lease, or time-share a computing system with the actual computer located at a central site, and use batch processing to execute students programs. This is the arrangement which exists in Baltimore City, Baltimore County, Carroll County, and Harford County, Maryland schools at the present time. It would seem then that studying the various effects of managing such a system would be beneficial to such systems in facing future decisions regarding expansion of computer facilities.

Related Studies

Many proponents of the teaching of computer programming claim that skills learned in such a course would carry over into other courses, such as mathematics. It is their belief that the skills developed in writing computer programs would transfer to solving mathematical problems.

This section will review literature related to the educational uses of computer programming as a course in aiding development of computational and problem-solving skills of mathematics students. It also includes some mention of recent studies relating the pertinent learning theory and computer programming. The first part deals with the effects on student achievement in mathematics of a course in computer programming.

Sherrill, (1973), conducted a study investigating the effects of different presentations of mathematical word problems upon the achievement of tenth grade mathematics students. Although his investigation was limited to the style of presentation of the word problem in the student's text, he nevertheless made some points pertinent to this investigation. Sherrill asserted that a part of the experience in problem solving is the development in the student of the ability to differentiate between the logical and illogical, the reasonable and the absurd. In constructing a computer program, especially one in which the student is first required to devise a flow chart solution, the student must be continually making decisions of this nature. He must learn to disregard the data which is not pertinent and to sort out the meaningful information. This information must then be organized as efficiently as possible. Sherrill further asserts that the student must not be protected from error, but rather encouraged to detect error, diagnose its cause, and demonstrate how to correct it; and, computer programming is an excellent vehicle for teaching these skills.

The results of Hatfield's study (1972) with seventh and eleventh grades in mathematics supported computer programming as a facilitator in certain aspects of mathematics instruction. Specifically, his findings indicate that even low-achievers can successfully learn to write and execute computer program solutions to problems. This is due in large part to the ability of computer programming to serve as a means by which the less able student can effectively organize his mathematical thinking. Hatfield suggests that his studies demonstrate that such higher order skills as problem-solving, independent inquiry, and generalizing can be enhanced with the design by the student of computer algorithms.

In fact, many mathematics educators have suggested that the activity of writing, processing, debugging, and studying the output of computer program solutions to algorithmic type problems should promote the development of mathematical concepts and principles, computational skills, and problem-solving abilities of the student. Kemeny, (1970), summed up this view in stating

> I feel that the right attitude is to teach them the algorithms in principle and then the right way to do the algorithm in practice is to program it for a computer. ... If a student succeeds in this he will have a depth of understanding of the problem which will be much greater than anything he has previously experienced.

A study of the effects of a computer programming experience on the problem solving abilities of eighth grade students was conducted by Foster (1972). He had students in an experimental group solve problems by constructing flow charts and writing computer programs. Students in a control group solved the same problems using paper-and-pencil.

Foster concluded that computer programming and flow charting of problem solutions tends to support the development of selected problem-solving behaviors. Similar findings were reported by Wilkinson, (1972), in a study of the effects on students' mathematical reasoning abilities of writing computer program solutions to problems. He found evidence of a significant increase in mathematical reasoning abilities among his students as a result of instruction in logical flowcharting. Wilkinson concluded that a course in computer programming is feasible as an adjunct of the comprehensive high school mathematics program, and that it would enhance students' abilities to develop reasoning abilities.

Thesing (1971) studied an approach to using a computer in the teaching of elementary calculus at the college level. The students in the study were freshmen calculus students who were divided into two groups, the experimental and the control. Those students in the control group completed their assignments using the usual pencil and paper method. The experimental group solved problems by writing computer program solutions, after receiving instruction in programming language. The significant finding was that one of the experimental groups, having lower ability and older mean age than the control group, gave evidence of improved achievement on topics to which the computer was applied.

A study was conducted in the Dearborn, Michigan, Public School system to determine if students who use a computer programming language in learning mathematics attain higher achievement levels, (Ronan, 1971). This report stated that higher achievement levels were attained by those who used a computer, in certain topics. Significant differences were found among those studying logarithms, in favor of the computer users, and in general, there was overall growth in understanding mathematical concepts, development of mathematical skills, and ability to perform mathematical problem solving. Those students who did not use a computer showed no significant growth in any of the above areas. Furthermore, the computer users showed significant differences in logical and reasoning abilities over those who did not use a computer.

Holoien (1970) conducted a study with the calculus students in a four year college, comparing a group using a computer to a group not using a computer. He found no significant differences between the groups on an attitude measure. However, significant differences were found in favor of those students among the lower ability subset who were in the computer group. Holoien concluded that a computer seemed helpful in learning calculus, especially among those students of lower ability. On the other hand, Fiedler (1969) found contradictory results to those of Holoien. Fiedler tested students in calculus with and without the use of a computer. One group did their home assignments by pencil and paper method, and the other wrote computer programs, where possible, in addition to the paper and pencil method. No significant differences were found at the 0.10 level. Fiedler concluded that the students learned the concepts studied just as well not using a computer as by using a computer.

One study reviewed was conducted at the high school level among students in trigonometry and physics. The author studied the effects of various arrangements of combining a course in computer programming with a course in trigonometry and one in physics. Results showed that one class which received 15 weeks of trigonometry combined with the writing of computer programs and use of flow charts performed significantly better, (Wallace, 1968). He concluded that the use of flow charts and algorithmic methods in teaching mathematics appears to fortify conventional teaching methods, with the result that higher learning rates are attained.

Summary

> ... since all professionals will be radically affected by the computer, all students will have to learn how it works and what it can do ... by using computers as data solving tools in such subjects as mathematics, physics, and economics.
> Bright (1967)

Certainly Bright's point concerning the impact of the computer on the professions, the arts, and the sciences is well taken. However, learning how to write a computer program and how a computer operates merely for the sake of being able to apply such knowledge is no more justifiable than learning how an auto works. Millions of people use autos everyday and most of then do not know how one works. Similarly, one need not learn to write programs or operate a computer to apply the information received from a computer. However, if we consider the strong case made by some of these studies reviewed for enhancement of learning through understanding and use of computer programming we can see the usefulness of such a course in the curriculum. The literature reviewed in the preceding pages seems to indicate that learning to write computer programs in a computer language and applying such ability in an instructional setting does have some positive effects on certain aspects of learning. There is evidence to support the notion that writing an algorithmic solution to a problem in a computer programming language does enhance the students' problem solving ability, (Sherrill, 1973; Hatfield, 1972; Foster, 1972; and Ronan, 1971).

The Dearborn study, (Ronan, 1971), gave substantive evidence to the enhancement of growth in mathematical concepts, skills, and problem-solving ability through a course in computer programming. Wilkinson's study (1972) indicates that the computer programming course aided in developing the students' reasoning abilities. These studies all support the supposition that a course in computer programming included in the curriculum would contribute positively toward certain aspects of student learning.

Related Studies-Learning Theory

The studies reviewed herein are classified in two groups, those related to various aspects of problem-solving, and those related to feedback schedules.

Sackman (1970) summarizes most of the findings of experimental psychology with relation to problem-solving, and applies these findings to problem-solving with the computer. He categorizes the findings into the classifications masses vs. spaced problem-solving, part vs. whole learning, and insight vs. trial and error. The thrust of Sackman's studies is the comparison of on-line vs. off-line program processing. His findings are mentioned in this report because they represent the single most significant contribution to the study of computer programming and problem-solving. Sackman states

> The data on massed vs. spaced (or distributed) learning suggests a useful analogy to distinguish between online and offline learning. The analogy is suggestive insofar as massed learning, as in time-sharing, represents more intensive and continuous problem-solving over longer blocks of time, in contrast to distributed learning, which, as in batch, represents shorter sessions with longer intervals between sessions. From another point of

> view, the contrast is between interrupted (batch) learning vs. noninterrupted (time-sharing) learning. The analogy breaks down with respect to subject feedback in that feedback is almost always immediate or soon after each trial in the problem-solving literature, whereas feedback is delayed by machine turn-around time in batch problem-solving.

He goes on to present the following table listing the comparative advantages of massed vs spaced problem-solving.

Advantages of Massed vs Spaced Learning

Advantages of Massed Effort	Advantages of Spaced Effort
Less warmup required.	Less fatigue and boredom.
Better for short tasks directly learned to completion.	Better for longer, but routine tasks.
Less forgetting between sessions.	Fewer interference effects and "mental ruts".
Better for tasks requiring much exploration.	Fewer trials to reach performance criterion.
Better for complex tasks.	Less dependence on immediate memory, more on long-term memory.

Sackman found that four out of five of the advantages listed under the Massed Effort column were demonstrated in the studies concerning time-sharing: less warmup, less forgetting between sessions, and a tendency toward better performance in exploratory and complex tasks. He relates the items in the Spaced Effort column to batch processing. It would seem that routine but time consuming tasks are an advantage of batch processing. Intermittent problem-solving might create fewer interference effects and "mental ruts". Most important is the fourth entry, fewer trials to performance criterion. The literature and studies Sackman reviewed all seemed to bear out less computer time and fewer program runs under batch to reach performance criterion. Finally, batch appears to facilitate the use of long-term memory in problem-solving rather than short term memory.

The problem-solving literature, according to Sackman's reviews, show the less sophisticated subjects doing better (fewer trials) when long tasks are broken down into smaller, more manageable pieces. The more sophisticated subjects seem to tackle larger parts or the entire task at one time and tend to require fewer trials to reach performance criterion. Sackman states, "The trend is toward larger parts and more 'gestalt' conceptualization with increasing learning, skill, and experience". Generally, the studies indicate that each individual should begin a task with the largest unit he can effectively handle in terms of his skill and experience. Also, this optimum unit tends to grow in response to feedback and progressive learning, generally increasing with successful experience. In a study by Gold (1967) the on-line mode (time-sharing) of processing was found to make it easier for the user to proceed in relatively small steps, thereby enabling the individual to set as an immediate task only that portion he felt capable of successfully completing. On the other hand, the batch mode seemed to cause the individual to gamble "all on one throw of the dice", that is, on one entire run of his program. Thus, the less sophisticated user was compelled to tackle as much of the task (the whole thing) as the more sophisticated one. Gold referred to the less sophisticated user as "marginal" users, and found that they tended to fare better under time-sharing processing mode, with fewer dropouts than under batch. In the Gold study, the time-sharing mode provided a more favorable learning climate for marginal users, gave immediate feedback, and was more "understanding" of human failure. Thus Gold's study would seem to bear out the fact that less sophisticated students in a course in computer programming do better under part-learning (time-sharing), while the more sophisticated learner functions well under whole (batch) learning.

Sackman discusses the issues of insight vs. trial-and-error. He defines insightful problem-solving as a kind of sudden learning with marked improvement in performance, a higher order grasping of problem relationships, an internal rehearsal of a successful solution. He refers to this as the "aha experience". On the other hand, trial-and-error problem-solving is a more gradual type of learning, marked by quasi-random exploratory behavior, many empirical trials, and by progressive reinforcement of the more successful responses until criterion is reached. Gestalt psychologists stress insightful problem-solving; behaviorists lean toward the more systematic trial-and-error method. Sackman found that neither method is actually ruled out completely in favor of the other, but rather both interact regularly in the successful completion of problem-solving tasks. If any trend was found, it would appear that trial-and-error was more fruitfully applied to tasks that cannot be conceived as a whole without extensive piecemeal feedback. On the other hand, insightful learning occurs whenever sufficient data exists for the formation of an "apperceptive mass", thus permitting the conceptualization and internal rehearsal of a solution. Gold's study (1967) seemed to demonstrate a "cyclical equilibrium" between trial-and-error and insightful problem-solving. He found that the first trial run in a time-sharing mode resulted in an improvement that averaged three times the improvement of succeeding trial runs of the same program solution, during the same console session, or sitting. Reasons given for this were generally that the students detected errors and corrected them, or attempted different factors, decision rules, or introduced new constants into the solution. Gold claims his findings represent the difference between insightful learning (the first trial run) and trial-and-error learning (subsequent trial runs). The first trial run resulted in changes to the problem solution which were an outgrowth of insightful, or gestalt-type conceptualizations. Subsequent trial runs were trial-and-error learning consisting of error corrections of the main theme. Thus, intervals between console sessions were spent to develop insightful strategies toward solving the problem, while sessions spent on the console were for the most part trial-and-error corrections of the program resulting from such strategies. Time-sharing mode processing leads to more exploratory trial-and-error learning which provides a richer data base for insightful thinking through the problem solution while away from the console. Gold's study involved two groups of students, 33 working under the time-sharing mode, 33 working under the batch mode. The task was an open-end computer simulation of the construction industry, in which each student was an independent builder whose objective was to maximize profit by improving decision making and optimizing available parameters. The time-sharing group averaged $110 extra profit in the first interaction in a computer session as opposed to $90 extra profit for the first batch run. Subsequent trials during the same console period added another $30 to the profit attained by the time-sharing group, thus widening the gap even further. Thus Gold concludes that the time-sharing mode aided insightful learning, as in the first trial run and trial-and-error learning, as in subsequent trials, more than did the batch mode.

In his study, Sackman mentions that delay of feedback under the batch processing mode could have been a factor. He makes no effort to determine in what way, or to what degree. In reviewing the literature, there seems to be little or no material available which investigates this important factor, with reference to computer programming. There is much material available on reinforcement schedules and feedback schedules in the psychological literature. Some of the findings are rather inconsistent. Newman (1974) found that a delay of feedback between 10 seconds and 24 hours yields an increment in retention when retention is measured five to seven days after feedback. Other literature reviewed showed relatively small effects of delay of feedback and that feedback interacts with other independent variables, (Fulton, 1969; Gay, 1972; Sassenrath, 1968). Newman's study involved undergraduates enrolled in a psychology course, assigned to read an article of about 3,700 words. A multiple choice test was administered, followed by one of four treatments: (1)no feedback, (2)immediate feedback, (3)one day feedback, and (4)seven day delayed feedback. A retention test was administered seven days after the feedback. No overall differences in the performance among the subjects was noted. A questionnaire completed by the students suggested that immediate feedback stimulated the most rereading. Thus, it would seem that such might be the case in a course in computer programming, where feedback of processed computer programs varies greatly, depending on several factors. It would also suggest that the feedback delay might be important to consider in relation to affective factors, such as attitude, motivation, and so on, rather than cognitive ones, such as achievement. Frye and Pack (1969) compared time-sharing, quick batch, and slow batch processing modes. Quick batch referred to processed computer programs returned one day later. Each student was to program as many as possible of 300 math and science problems. The three groups were randomly assigned to a mode of processing. A test administered at the end of four weeks covering the language learned (BASIC) showed no significant differences in learning, but significantly more problems solved under either batch mode. Also, the cost factor was studied, with the following results:

Time-sharing	$21.29 per problem
Quick batch	$6.21 per problem
Slow batch	$6.05 per problem

A questionnaire indicated that the students preferred time-sharing.

Several other studies were concerned with the relationship between computer programming and the learning of mathematics. Meserve (1968) stated that "students who acquire a working introduction to algorithmic languages while in high school gain greater insight into mathematics". Morgan (1968) found that general education students could enhance their mathematical competencies when content is integrated with computer based applications. His subjects expressed an attitude favoring the inclusion in the general mathematics curriculum of some computer programming instruction. Ingle (1973) found that students' perceptions of the computer were largely due to ignorance of its function and operations. This was especially true among students who had been exposed to Computer Assisted Instruction. These students tended to attribute to the computer characteristics of creative intelligence and high-expertise in certain areas.

Milner (1972) conducted an intensive study of the effects on teaching computer programming on performance in mathematics. He found the following:

(1) Computer programming is an effective learning resource in terms of both cognitive and affective considerations.

(2) The method of teaching computer programming is less important than the definition of suitable tasks and the preparation for them.

(3) Computer programming, in his study, facilitated the acquisition of certain problem-solving behaviors.

(4) Some students whose motivation was previously relatively low were "turned-on" by the computer programming course.

Pellegrino (1971) also found a significant effect upon affective factors by a course in computer programming. His students exhibited a significant positive shift in attitude toward the computer and education in general.

Summary

The studies reviewed in the preceding pages reveal some interesting data concerning the computer as an educational device. First, the computer is an important tool which can be used effectively as a teaching device. This, of course, includes not only its usual role as a facilitator in Computer Assisted Instruction but also as a facilitator of certain desirable learning traits. The computer course offered to students in several studies was found to aid the students in problem-solving in mathematics and science courses. Second, the computer as an education device in the classroom can enhance interest and motivation. This was especially true among lower ability groups in some studies reviewed. Third, there is some question raised among proponents of the teaching of computer programming as to the most desirable mode of presenting such a course.

Questions were raised concerning the most efficient mode of operation, time-sharing or batch. There is substantial evidence in favor of each as the preferred mode of operation. Given the batch mode of processing, which arrangement for executing programs is better, quick batch or slow batch, or is there any difference? Is there a difference in achievement or motivation attributable to turn-around time? These last two questions will be considered in this paper.

Chapter 3 presents the description of the experimental design used in this investigation.

CHAPTER III

THE EXPERIMENTAL DESIGN

The Problem

The purpose of this study was to determine the effects of frequency of direct association with the computer and turn-around time on student attitude and achievement in a beginning course in computer programming.

Specifically, this study investigated the following questions:

1. Is there any effect on student attitude and achievement due to the frequency with which the student comes into direct association with the computer in a beginning course in computer programming?
2. Is there any effect on student attitude and achievement due to the time lapse between submittal of a program for processing and return of the processed program in a beginning course in computer programming?

Two hypotheses were tested with respect to achievement, and two with respect to attitude.

Hypotheses

1. There is no difference in FORTRAN programming competency as measured by the final examination between students who are in direct association with the computer less than one

hour per term, those in direct association approximately one hour per week per term, and those in direct association three hours per week per term.

2. There is no difference in FORTRAN programming competency as measured by the final examination between students who receive their printouts on the same day in which they are submitted, those who receive their printouts about one day later, and those who receive their printouts about one week later.

Two treatments were applied: frequency of direct association with the computer (F); and time delay between submittal of the program and receipt of the processed printout (T). Each treatment consisted of three levels. Treatment F, frequency, consisted of the following three levels:

H0 - one hour per term

H1 - one hour per week per term

H3 - three hours per week per term

Due to the class scheduling program under which the school operated, the term "hour" in this study actually referred to a class session of 50 minutes duration, the length of one standard period during a regular school day. The second treatment, T, consisted of these three levels:

PI - printouts returned on the same day as program submitted

PD - printouts returned in 24 to 48 hours of time of submittal; (about one day)

PW - printouts returned during a period of time lapse not less than 7 calendar days, nor more than 8 calendar days; (about one week)

The two hypotheses with respect to achievement were analyzed using analysis of covariance on a 3 x 3 design. The final examination was the instrument used to determine the criterion, and the covariate was programming ability, as measured by an aptitude test. The hypotheses tested with respect to attitude were:

3. There is no difference in attitude toward the computer as measured by the Computer Attitude Instrument between students who are in direct association with the computer less than one hour per term, those in direct association one hour per week per term, and those in direct association three hours per week per term.
4. There is no difference in attitude toward the computer as measured by the Computer Attitude Instrument between students who receive their printouts on the same day they are submitted, those who receive their printouts about one day after submittal, and those who receive their printouts about one week after submittal.

The same treatments were applied with respect to attitude as were applied with respect to achievement, each at the same three levels. The analysis of covariance was used to analyze the results, with the aptitude examination score as covariate and the Computer Attitude Instrument score as criterion.

Definition of Terms

1. Batch Processing

   This refers to the sequential programming (or executing or running) of a queue of programs on a periodic basis. Generally, the programs are presented in decks of cards. The cards may be keypunched or pencil marked (also called "mark sense" cards). In the IBM 1130 system, such cards are placed in a machine known as a card reader.

2. Turn-around Time

   This refers to the time lapse between submittal of the prepared program deck to the computer for processing and return of the processed program, in this case, in the form of a printout.

3. Mark Sense Cards

   These are cards of approximately the same size and shape as the usual keypunch cards, but are printed on one side with letters, numbers, and symbols of FORTRAN programming language, arranged in 18 columns. The user codes the program on the card by marking the appropriate character in each column with a dark pencil.

Evaluative Instruments

1. The Programming Aptitude Instrument (ATPP)

   The IBM Aptitude Test for Programming Personnel (IBM, 1964) is a timed test and required two fifty minute sessions each to administer. The test was given to the students during the first double period session of the course and administered by the investigator, following precisely the directions supplied with the test. This particular instrument had been used for a number of years by the IBM Corporation to identify prospective employees having an aptitude for computer programming. The test manual reports the reliability and validity estimates for this test. Using the scores of one hundred forty-four junior college students in a test-retest situation, a reliability of .88 was found. IBM Corporation obtained the validity of prediction of success in training by correlating the ATPP scores with the final grades in three IBM education center classes. The average correlations (Fisher z' used for averages) were .45, .56, and .64, which were all significant beyond the .01 level.

2. The Computer Attitude Instrument

   A Likert-type attitude instrument was constructed to measure the attitude of the students toward the computer with respect to aspects of life. The aspects of life were:

   a. Education.

   b. Science and industry.

   c. Society in general.

   The instrument consisted of twelve statements, each reflecting an attitude or opinion concerning the computer in relation to one of the aspects of life. Each statement was followed by a rating of five intervals, ranging from Strongly Agree to No Opinion to Strongly Disagree. The twelve statements included both positive and negative feelings, divided equally among the three aspects of life. In writing the statements, several pairs of adjectives were used, each pair consisting of bi-polar opposite words. These pairs were:

useful	-	useless
safe	-	dangerous
harmless	-	harmful (or evil)
helpful	-	no good
important	-	worthless

The positive and negative statements were alternated. The attitude instrument was administered to all participants at the end of the course, immediately prior to the final examination. Reliability estimates were determined and are presented in Chapter IV. The attitude instrument appears in Appendix B.

3. The Final Examination

The instrument used to determine student achievement in the computer programming course was a test devised by the teachers of computer programming at the Baltimore Polytechnic Institute. It is a multiple-choice test, designed to be administered in one 50 minute period. There were 25 questions with five choices per question. Reliability estimates were determined after administration of the final examination, and the results appear in Chapter IV. The test is in Appendix D. Validity was determined by a questionnaire completed by a panel of experts in the field of computer education in the Baltimore City Public School System. The questionnaire and results appear in Appendix E and Appendix F, respectively.

Description of Sample

The sample selected was the set of all tenth grade students enrolled in a beginning course in computer programming at the Baltimore Polytechnic Institute during the Spring, 1975, term. The course consisted of combined Geometry and Computer Programming, FORTRAN IV, and classes met six periods per week, including one double period. Three periods were spent on the Geometry, and three periods on computer programming. Programs were run using the IBM 1130 computing system. This sample represented the set of all tenth grade students who enroll in a course in beginning computer programming at BPI.

Due to the method of assigning students to classes at BPI, it was not possible to randomly select students for the various treatment applications. In order to minimize any negative effects attributable to this lack of randomization, the following procedures were instituted.

1. The course was designed to be an individualized study course with the student working at his own pace. The text selected for use was Communicating With The Computer, (Jacobs, French, Moulds, Schuchman, 1974), which was specifically designed for such a computer programming course. The students were given written assignments, readings, and problems to be solved by computer programs. Instructions to the teachers

recommended use of the text and a minimal amount of lecture and discussion. The course outline and suggested assignments are found in Appendix A.

2. Classes were assigned to treatment blocks based upon class means and variances as found in Table 2. A Chi square goodness-of-fit test was run on the nine class means to compare each class score on the Iowa Test of Basic Skills, Mathematics Level 9, to a normal curve. The Chi square statistic for each class also appears in Table 2. None of the Chi square scores showed any significant difference between the class scores and the normal distribution (critical Chi square $_{(.95,2)}$ = 5.99). Hartley's Fmax Test was applied to determine homogeneity of variances. The value obtained was F = 2.45 which was not significant (critical $Fmax_{(9,27)}$ = 3.519). Having established that, based on the Iowa Test Arithmetic scores, each group did not significantly depart from a normal distribution, the assignment to treatments was carried out. An attempt was made to distribute the nine classes across the treatments in such a manner that no one level of any treatment would have a significantly higher mean on the Iowa Test Arithmetic scores than any other treatment level group. However, only six teachers were assigned to teach the nine classes, one teacher being assigned to teach three of the classes, another being assigned two classes. The decision was made to meet with the teachers and have them select the treatment combinations with which they felt comfortable.

There were nine treatment combinations possible, using the treatment levels:

HO - One hour per semester in laboratory

H1 - One hour per week in laboratory

H3 - Three hours per week in laboratory

PI - Immediate turn-around time (less than one day)

PD - About one day turn-around time

PW - About one week turn-around time

The teacher with three classes agreed to assign his classes so that he would have one and only one class in each treatment level. His classes are those on the diagonal upper left to lower right in Table 2. Some of the remaining teachers expressed definite preferences for certain treatment level applications, (such as one teacher who preferred never meeting his class in the computer laboratory). The remaining classes were placed in treatment combination categories according to teacher preference, with one restriction. An effort was made, based on the data available at the time, to prevent any one treatment level being assigned predominantly better or poorer classes, based on the arithmetic scores. Final assignment of classes to treatment combinations (cells) is displayed in Table 2. The school designation for these nine classes was N1, N2, N3, ..., N9. Each of the classes consisted of at least 30 students, with one class having 36 enrolled. The decision was made to maintain an equal number of subjects in each cell of the analysis. This was accomplished by using in the analysis of the experiment only the first 28 subjects in each cell on whom complete data was collected. The data included the Iowa Test Arithmetic score, the Aptitude Test score, the final examination score, and the Attitude Test score. Due to incomplete school records and student absence, it was possible to control cell size in this manner.

TABLE 2

Means, Standard Deviations, and Chi Square Values
Iowa Test of Basic Skills - Form 9
Mathematics

Treatment F (Frequency)

Treatment T Time			H0		H1		H3	Mean
PI	M.	9.289		9.321		10.189	9.600
	S. D.	1.365		1.131		1.244	
	X^2	1.300		1.02		2.430	
	Class	N8	Class	N4	Class	N7	
PD	M.	10.768		9.282		8.993	9.681
	S.D.	.882		1.400		1.298	
	X^2	1.310		3.250		.880	
	Class	N1	Class	N6	Class	N9	
PW	M.	9.221		8.632		10.279	9.444
	S.D.	1.193		1.269		.961	
	X^2	4.61		.736		.680	
	Class	N3	Class	N5	Class	N2	
Mean		9.759		9.145		9.820	9.575

H0 - One hour per semester

H1 - One hour per week

H3 - Three hours per week

PI - Immediate turn-around time

PD - About one day turn-around time

PW - About one week turn-around time

Procedures for Analyses

Analysis of covariance was chosen as the method of evaluating the data obtained from application of the treatments. The student scores on the Aptitude Test for Programming Personnel was selected as the covariate, for both achievement and attitude. The Aptitude Test (ATPP) was administered to the students at the first double period class meeting of the course. A course outline and assignment sheet was provided for each teacher at the beginning of the course, together with instructions on treatment applications to their classes. They were asked to report any deviations from treatments. All students were given an introductory lesson to programming and the computer in general, and all were taken to the computer laboratory for a "tour". Thereafter, the treatment applications were strictly observed, except as noted under limitations to the study in Chapter V. The students were given instructions in the use of the keypunch machine and of mark sense cards for preparation of their program decks. There was no restriction placed on the students as to which method of preparing their decks they could use. The computer laboratory normally contains four keypunch machines, but for those students who were assigned to the treatment groups which never attended the computer laboratory one machine was removed and placed in a classroom. The students who were assigned to the computer laboratory one hour per week were allowed to use the keypunch machines in the laboratory but only during that one hour period. The remaining students, those assigned to the laboratory for three hours per week, were issued special passes, and allowed to use the computer laboratory during class time and before and after school as well. Their use of the facility was monitored by diligent checking of passes of all students found in the laboratory outside of regular school hours. In order to control the turn-around time for processing programs, each student was assigned a student number which was to be placed in a conspicuous place on the first card of each deck submitted for processing. The numbers consisted of six digits indicating the student's class, roll number, and a code representing his turn-around time treatment. Program decks were checked by the computer operators before processing. The treatments were continued for 15 weeks, the final week of class being reserved for administration of the various instruments. The students were asked to complete a questionnaire related to the computer programming course and this study. The questionnaire and results appear in Appendix G. The final examination was administered to all students during a special examination time during the last week of classes. Results are reported in Chapter IV.

CHAPTER IV

Results of the Experiment

The results of the statistical hypotheses were analyzed using the MANOVA program on file at the University of Maryland Computer Science Center. Reliability estimates for the attitude instrument and final examination were obtained using item analysis programs included in the Scientific Subroutine Package (SSP) supplied by the IBM Corporation.

During the first double-period class meeting, the Aptitude Test for Programmer Personnel was administered to each of the nine groups. The cell means and standard deviations are displayed in Table 3. A grand mean of 42.401 was found for all 252 subjects. Scores ranged from 11 to 75 based on a maximum possible score of 95. There were 28 subjects per cell. Test scoring instructions included a penalty for guessing, which explains how a low score of 11 could be possible. IBM reported a mean of 51.4 and standard deviation of 15.9 attained by the population upon which the test was normed. The mean age for the norm group was 30.2 years, and mean education level was 14.3 years. The group in this study had a mean age of 15.7 years and all had 10 years of formal education. In order to test the assumption of homogeneity of variance, a Hartley's Fmax statistic was computed. The obtained result of F = 2.383 was not significant, (critical $F_{(.95,27)} = 3.52$). The assumption of homogeneity of variance was not rejected.

An analysis of the Iowa Test Mathematics scores showed F = 9.342, which was highly significant (critical $F_{(.95,8,243)} = 1.938$). Although this indicates significant differences among these means, the decision was made to proceed with the analysis of covariance using the Aptitude Test scores as covariate, as originally designed. This decision was made after the analysis of covariance was carried out using both the Aptitude Test scores and the Iowa Test scores as covariates. The conclusion was that using the Iowa Test as a second covariate added no significant information to the study.

TABLE 3

Means and Standard Deviations
for the Aptitude Test for Programmer Personnel

		Frequency			
		H0	H1	H3	Treatment Mean
Time	PI	M: 48.821 S: 8.849	39.821 9.813	48.893 10.450	45.845
	PD	M: 50.214 S: 10.411	41.679 11.941	37.286 7.736	43.060
	PW	M: 37.107 S: 11.902	33.536 10.507	44.250 10.083	38.298
Treatment Mean		45.381	38.345	43.476	42.401

Frequency Levels:

H0 - One hour per term in direct association

H1 - One hour per week in direct association

H3 - Three hours per week in direct association

Time Levels:

PI - Less than one day turn-around time

PD - One day turn-around time

PW - One week turn-around time

Achievement

The final examination was administered to all students simultaneously during the last week of classes, at a specially designated time. The data obtained was analyzed using a 3 x 3 analysis of covariance design. The Aptitude Test for Programmer Personnel scores were used as the covariate.

Cell means and standard deviations of the unadjusted scores on the final examination are displayed in Table 4. The final examination was a twenty-five question, multiple-choice test with five choices per question. The highest score attained was twenty-two, while the lowest score was two. The grand mean obtained was 12.452, the standard deviation was 2.67, and the reliability estimate was .77 with N = 252. The assumption of homogeneity of variance was tested using Hartley's Fmax statistic. The computed Hartley's Fmax was F = 3.275 (critical $F_{(.95,9,27)}$ = 3.52). Thus the assumption of homogeneity of variance was supported.

One of the assumptions underlying the analysis of covariance is that of homogeneity of regression. This assumption was tested using the computed data displayed in Table 5.

TABLE 4

Means and Standard Deviations
Unadjusted Scores
Final Examination

		Frequency H0	Frequency H1	Frequency H3	Treatment Mean
Time	PI	M: 14.000 S: 2.776	11.679 2.161	14.786 2.807	13.488
Time	PD	M: 16.643 S: 1.810	12.500 3.191	10.071 3.005	13.071
Time	PW	M: 7.857 S: 2.649	8.250 2.351	16.286 3.276	10.798
Treatment Mean		12.833	10.810	13.714	12.452

TABLE 5

Summary Table for Test of Homogeneity of Regression - Achievement

Source	df	Mean Square	F
Among slopes	8	8.140	1.213
Deviations	234	6.713	
(Critical $F_{(.95,8,\infty)}$ = 1.94)			

The assumption of homogeneity of regression was not rejected, F = 1.213 (critical $F_{(.95,8,\infty)}$ = 1.94). The analysis of covariance was carried out, and the results are displayed in Table 6. Significant differences were noted for the treatment effects, Time and Frequency, and the interaction effect. Decisions concerning the hypotheses related to achievement are based on the findings in Table 6.

TABLE 6

Summary Table
Analysis of Covariance
Achievement

Source	df	MS	F	P less than
F	2	128.053	18.941	.001
T	2	105.549	15.612	.001
TF	4	299.168	44.252	.001
Regression	1	146.450		
Within Cells	242	6.761		

Hypothesis 1: The hypothesis concerning no difference in achievement between students in direct association with the computer at varying frequencies was rejected, ($F_{(2,242)} = 18.941$, $p < .01$). Thus, the decision was made to conduct some post-hoc analysis to locate where the differences were to be found. The method attributed to Tukey (Winer, 1971) was used to examine comparisons among all pairs of adjusted means. The adjusted means for all cells, and the treatment means, are displayed in Table 7.

Decisions made concerning the effects of the levels of treatment F (frequency of direct association) were based on the analysis summarized in Table 8.

TABLE 7

Means and Standard Deviations
Adjusted Scores
Final Examination
ATPP as Covariate

				Frequency		
			H0	H1	H3	Treatment Mean
Time	PI	M:	13.514	11.874	14.295	13.228
		S.D.	2.659	2.070	2.689	
	PD	M:	16.052	12.555	10.458	13.022
		S.D.	1.734	3.058	2.879	
	PW	M:	8.258	8.920	16.146	11.108
		S.D.	2.538	2.252	3.138	
Treatment Mean			12.608	11.116	13.633	12.453

TABLE 8

Summary Table for Pairwise Comparisons
Among Adjusted Means on Final Examination
Frequency Treatment

Pair	Difference
H0 - H1	1.492*
H0 - H3	-1.025*
H1 - H3	-2.517*

Adjusted MS_e = 6.822

*Critical Difference = .943

The critical difference of .943 is exceeded by the differences between all pairs of means. Students in direct association with the computer one hour per week scored significantly lower than students in direct association three hours per week, and significantly lower than the group in direct association one hour per term. Those students in direct association three hours per week scored significantly higher than students in either of the other two treatment groups.

Hypothesis 2: The hypothesis concerning no difference in achievement between students experiencing the three treatment T (turn-around time) levels was not supported by the findings of this experiment, ($F_{(2,242)}$ = 15.612, $p < .01$). Again the decision was made to conduct post-hoc data analysis tests using the Tukey method on all pairwise comparisons. Decisions made concerning the levels of treatment T (turn-around time) were based on the analysis summarized in Table 9.

The critical difference of .943 is exceeded by the differences between two pairs of means. The group experiencing a turn-around time of one week scored significantly lower than the group receiving their printouts immediately (less than one day), and significantly lower than the group receiving their printouts in about one day.

TABLE 9

Summary Table for Pairwise Comparisons
Among Adjusted Means on Final Examination
Time Treatment

Pair	Difference
PI - PD	.206
PI - PW	2.120*
PD - PW	1.914*

Adjusted MS_e = 6.822

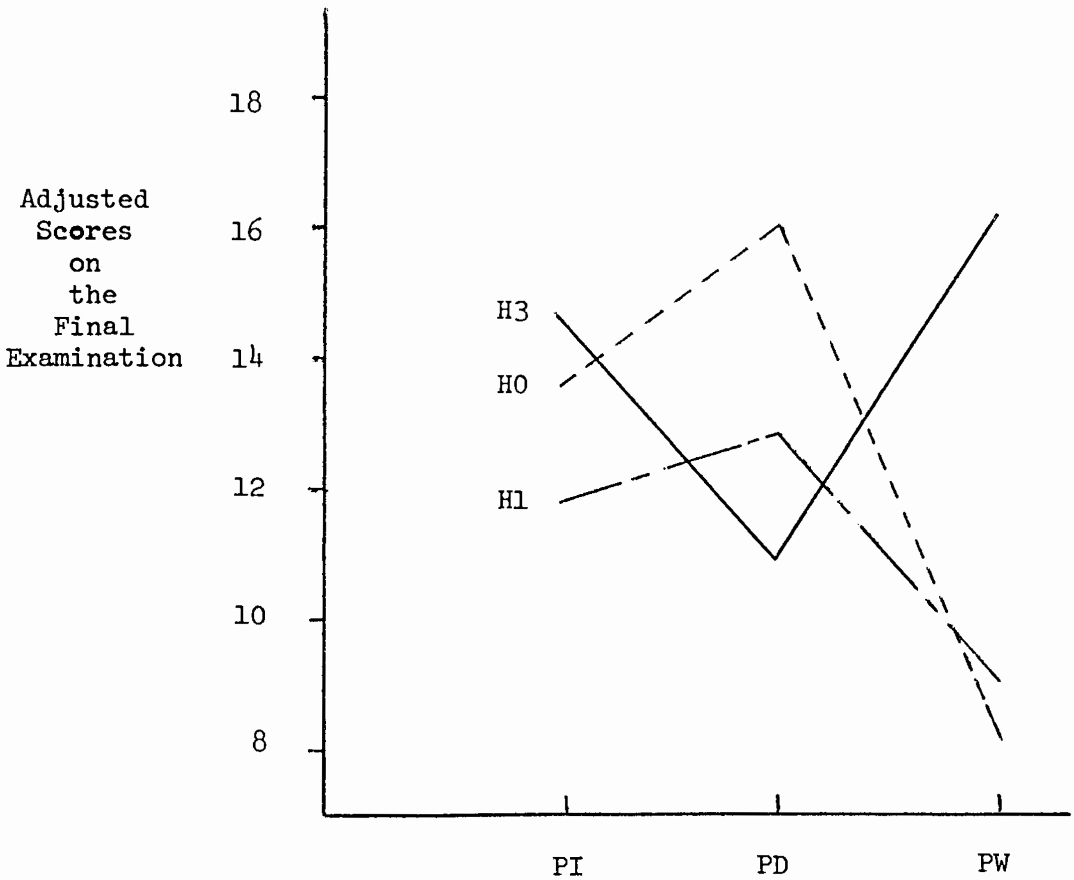
*Critical Difference = .943

Interaction - Achievement

There clearly are significant differences in achievement due to interaction effects. Strategies concerning the post-hoc investigation of significant effects due to unusual combinations of frequency and turn-around time were guided by the interaction diagram in Figure 1.

Figure 1

TF Interaction Effects
Achievement



Code H0 - One hour/term PI - Less than 24 hour turn-around time

H1 - One hour/week PD - About one day turn-around time

H3 - Three hours/wk PW - About one week turn-around time

The graph shows that treatment H3 has a deleterious effect on performance on the final examination in conjunction with treatment PD. Using the Scheffé procedure for contrasting cell means, this interaction effect was tested, and decisions were based on the analysis summarized in Table 10.

TABLE 10

Summary of Scheffé Tests for
Time-Frequency Interaction Effects
on Achievement

Source	df	Mean Square	F
PI. H3 - PI. H1 vs. PD. H3 - PD. H1	1	142.886	20.945*
PD. H3 - PD. H1 vs. PW. H3 - PW. H1	1	608.428	89.186*
PD. H0 - PD. H1 vs. PW. H0 - PW. H1	1	121.081	17.749*

Adjusted $MS_e = 6.822$

*Critical value $4 \times F_{(.95,4,\infty)} = 9.6$

The first two comparisons in Table 10 indicate significant interaction occurred involving treatment H3 in conjunction with treatment PD. Students in direct association with the computer three hours per week scored highest of all frequency treatment students on the final examination, except the group which also had a turn-around time of one day. The combination of three hours in direct association and one day turn-around time produced significantly disordinal interaction. The group under that treatment combination scored lowest of the three groups under one day turn-around time.

Rather unusual interaction results were found among all three frequency treatment groups under a one week turn-around time when compared to the same groups under one day turn-around time. The third comparison in Table 10 indicates the difference in performance on the final examination between the H0 and H1 groups under one day turn-around time was significantly greater than the difference between these two groups under one week turn-around time. Furthermore, the graph indicates this interaction to be disordinal.

The Tukey procedure for testing all pairwise comparisons among the nine treatment combinations was carried out in order to examine simple effects within treatment levels. Decisions based upon treatment combinations of interest are summarized in Table 11.

TABLE 11

Summary Table
Comparison of Certain Treatment Combinations

Treatment Level	Pair	Difference
PI	PI. H1 vs. PI. H0	1.640
	PI. H1 vs. PI. H3	2.421*
PD	PD. H0 vs. PD. H1	3.497*
	PD. H0 vs. PD. H3	5.594*
PW	PW. H3 vs. PW. H1	7.226*
	PW. H3 vs. PW. H0	7.888*

Adjusted MS_e = 6.822

*Critical Difference = 2.167

Under the PI treatment level (immediate turn-around time), students in the computer laboratory one hour per week scored lower on the final examination than students in either of the other two frequency groups, and significantly lower than those in the laboratory three hours per week. Students who were almost never in the computer laboratory scored significantly higher than the students who were in the laboratory more frequently among all students experiencing one day turn-around time.

The students experiencing the treatment combination three hours per week in the laboratory in conjunction with a one week turn-around time scored significantly higher on the final examination than either of the other two groups under one week turn-around time.

These finding indicate unusual results in achievement due to certain combinations of frequency of association and immediacy of feedback. Meeting in the computer laboratory three hours per week seemed to cause unusual results across the three levels of the feedback treatment.

Attitude

At the end of the course, the attitude instrument was administered to each group. The data obtained by the Likert-type attitude instrument was analyzed using a 3 x 3 analysis of covariance design. The Aptitude Test for Programmer Personnel was the covariate.

Cell means and standard deviations of the unadjusted scores are displayed in Table 12. Students expressed their degree of agreement with each of the twelve statements on a scale ranging from one to five. A rating of one was considered very negative, while a rating of five was considered very positive and a rating of three indicated no opinion. A non-response to an item was scored as a zero. The high score was 60, and the low score was 24. An item analysis was performed on the data using a statistical program which is part of the IBM Scientific Subroutine Package. This item analysis appears in Appendix C. The program also computes the reliability estimate using the odd-even method and the Spearman-Brown correction. The reliability estimate obtained was 0.79, with a mean of 43.274 and standard deviation of 6.384. In order to test for homogeneity of variance, an Fmax statistic was computed using the unadjusted scores. The maximum variance was 54.720, and the minimum variance was 16.988, yielding F = 3.22, which was not significant, (critical $F_{(.95,9,27)} = 3.519$). Hence it was assumed that there were no significant differences in the variances among the nine groups.

TABLE 12

Means and Standard Deviations
Unadjusted Scores
Attitude Instrument

Frequency

Time		HO	H1	H3	Treatment Mean
	PI	M: 44.893 S: 4.122	44.536 6.523	43.393 5.877	44.274
	PD	M: 44.714 S: 6.688	44.071 5.610	40.143 7.397	42.976
	PW	M: 41.107 S: 7.345	43.714 6.335	42.893 6.100	42.571
Treatment Mean		43.571	44.107	42.143	43.273

The next step in analyzing the data pertaining to attitude was to conduct the analysis of covariance using the ATPP as the covariate. One of the assumptions underlying the analysis of covariance involves the homogeneity of regression. This assumption was tested using the computed data displayed in Table 13.

TABLE 13

Summary Table for
Test of Homogeneity of Regression
Attitude

Source	df	Mean Square	F
Among slopes	8	36.760	.9459
Deviations	234	38.862	

Critical $F_{(.95,8,\infty)} = 1.94$

The obtained value F = .9459 was not significant. Hence, the assumption of homogeneity of regression was supported. The analysis of covariance was carried out, and the results are displayed in Table 14. The findings with respect to the hypotheses concerning attitude are based on these results.

TABLE 14

Summary Table
Analysis of Covariance
Attitude

Source	df	MS	F	P less than
F	2	121.475	3.131	.045
T	2	27.064	.698	.499
TF	4	34.482	.889	.471
Regression	1	232.555	5.995	
Within cell	242	38.793		

TABLE 15

Adjusted Means on Attitude Instrument
ATPP as Covariate

Frequency

			H0	H1	H3	Treatment Mean
T	PI	M	44.281	44.782	42.774	43.946
i m	PD	M	43.970	44.140	40.630	42.913
e	PW	M	41.612	44.559	42.717	42.963
Treatment Mean			43.288	44.494	42.040	43.274

Hypothesis 3: The hypothesis concerning no difference in attitude toward the computer between students in direct association with the computer at varying frequencies was rejected. The adjusted means are displayed in Table 15. The decision was made to conduct some post-hoc data analysis to determine where the differences were to be found. The method due to Tukey (Winer, 1971) was used to examine comparisons among all pairs of adjusted means. Conclusions concerning the effect of treatment F are based on the analysis summarized in Table 16.

TABLE 16

Summary Table for Pairwise Comparisons
Among Adjusted Means on Attitude
Frequency Treatment

Pairs	Differences
H0 - H1	- 1.206
H0 - H3	1.248
H1 - H3	2.454*

Adjusted MS_e = 39.142

*Critical Difference = 2.259

The critical difference was exceeded by one pair of means. Students in direct association with the computer one hour per week expressed a significantly more positive attitude than those students in direct association three hours per week. In fact, the students in the three hour per week treatment group expressed a less positive attitude than students in either of the other two groups.

Hypothesis 4: The hypothesis concerning no difference in attitude toward the computer between students experiencing the three levels of treatment T (turn-around time) was supported. Students receiving their printouts immediately, (less than one day) expressed the most positive attitude, but not significantly different from either of the other two groups.

Interaction - Attitude

The results of the 3 x 3 analysis of covariance indicate no significant differences in interaction due to unusual combinations of frequency of association and turn-around time. Further investigation reveals the highest adjusted mean (44.782) was attained by the group under interaction effect H1 vs PW (those in the lab one hour per week, but receiving their printouts about one week after submittal of programs). The lowest adjusted mean of 40.630 was attained by the group H3 vs PD (those in the lab three hours per week and having to wait one day for their printouts).

Summary

Hypothesis 1: There is no difference in FORTRAN programming ability as measured by the final examination between students who are in direct association with the computer less than one hour per term, those in direct association approximately one hour per week, and those in direct association three hours per week.

Conclusion: The results of this study did not support Hypothesis 1, therefore, the Ho was rejected.

Hypothesis 2: There is no difference in FORTRAN programming as measured by the final examination between students who receive their printouts on the same day in which they are submitted, those who receive their printouts about one day later, and those who receive their printouts about one week later.

Conclusion: The results of this study did not support Hypothesis 2, therefore, the Ho was rejected.

Hypothesis 3: There is no difference in attitude toward the computer as measured by the Computer Attitude Instrument between students who are in direct association with the computer less than one hour per term, those in direct association approximately one hour per week, and those in direct association three hours per week.

Conclusion: The results of this study did not support Hypothesis 3, therefore, the Ho was rejected.

Hypothesis 4: There is no difference in attitude toward the computer as measured by the Computer Attitude Instrument between students who receive their printouts on the same day in which they are submitted, those who receive their printouts about one day later, and those who receive their printouts about one week later.

Conclusion: The results of this study did support Hypothesis 4.

Significant interaction effects were noted with respect to achievement. Treatment PD had deleterious effects in conjunction with frequency treatment H3. No significant interaction effects were noted with respect to attitude.

Results of the Questionnaire

Near the end of the course, each student was asked to complete a questionnaire. The questionnaire had two objectives:

1. To confirm that the students in each treatment group had indeed received that treatment.
2. To gain information regarding operational characteristics of the course which would lead to better instruction.

All 252 students submitted a questionnaire, but several were incomplete.

The questionnaire appears in Appendix G. The results are summarized below.

Item 1. The students reported frequencies of direct association as shown in table 17. One group of eight students reported they were in the computer laboratory three hours per week rather than the one hour per week called for by the treatment.

TABLE 17

Summary Table
Student Report on Frequency Treatment

Treatment Reported		H0			H1			H3			
		H0	H1	H3	H0	H1	H3	H0	H1	H3	Total
Number reporting in each cell	PI	28	0	0	0	28	0	0	0	28	84
	PD	28	0	0	0	20	0	0	8	28	84
	PW	28	0	0	0	28	0	0	0	28	84
Total		84			76			92			252

Actual number of subjects in each cell = 28

Item 2. The students reported the turn-around times as shown in Table 18.

TABLE 18

Summary Table
Student Report on Time Treatment

Treatment Reported		PI			PD			PW			
		PI	PD	PW	PI	PD	PW	PI	PD	PW	Total
Number reporting in each cell	H0	28	3	0	0	25	0	0	0	28	84
	H1	27	0	1	1	28	3	0	0	24	84
	H3	26	0	0	2	28	0	0	0	28	84
Total		85			87			80			252

The data on remaining items are reported in terms of per cent of student answering.

Item 3. Number of times required to process a program until correct printout obtained.

One time 21.4%

Two times 63.5%

More than two times 15.1%

Item 4. Number of programs processed using punched cards.

None: 40%

All: 22.4%

At least one, but not all: 37.6%

Item 5. Number of programs processed using mark sense cards.

None: 46.1%

All: 34.6%

At least one, but not all: 19.3%

Item 6. Students using both mark sense and punch cards: 20.5%

Of these, 62.8% preferred punch cards.

Item 7. Preferred frequency of direct association.

Number of Hours	Per Cent Preferring
3 per week	84.6%
1 per week	9.5%
1 per term	5.8%

Item 8. Preferred turn-around time.

Turn-around Time	Per Cent Preferring
About one week	13%
About one day	28.9%
Less than one day	57.9%

Students were also asked to report on which aspects of the computer programming course they liked best, and which aspects they liked least.

Item 10. Aspects liked most:

Comments: Using computers

Using the keypunch machines

A different experience

It was interesting

Writing a successful program

Non-traditional classroom atmosphere

No written homework

It was different and challenging

Learning about a new field (or career)

Item 11. Aspects liked least.

Comments: Mark sense cards

Too rushed

Long wait for keypunch

Not enough keypunches

Course was boring

Broken keypunches

Computer shutdown too often

Time delay waiting for printouts

CHAPTER V

Summary and Conclusion

Statement of the Problem

The problem was to evaluate the effect of turn-around time and frequency of direct association with the computer on student attitude and achievement in a beginning course in FORTRAN IV programming at the secondary school level.

Summary

Two specific questions were investigated:

1. Is there any effect on student attitude and achievement due to the frequency with which the student comes into direct association with the computer in a beginning course in computer programming?
2. Is there any effect on student attitude and achievement due to the time lapse between submittal of a program for processing and return of the processed program in a beginning course in computer programming?

Two treatment factors were applied: T - turn-around time; and, F - frequency of direct association. Each existed at three levels. The three levels of T were:

PI - less than 24 hour turn-around time

PD - about one day turn-around time

PW - about one week turn-around time

The three levels of F were:

H0 - one hour of direct association per term

H1 - one hour of direct association per week

H3 - three hours of direct association per week

A 3 x 3 analysis of covariance design was used to evaluate the data obtained from application of the instruments. The students in the sample could not be randomly assigned due to the method of block scheduling used by the school. One of the assumptions of analysis of covariance is that the sample data being analyzed is a randomly selected representation of a normally distributed population. Therefore, in order to determine that treatment populations did not significantly depart from normal distribution a Chi square Goodness-of-Fit Test was run using the Iowa Test of Basic Skills, Math Form 9, as criterion. No significant departures from normality were found among the nine groups. Likewise, a Hartley's Fmax test supported the assumption of homogeneity of variance, another assumption of analysis of covariance.

The Aptitude Test for Programmer Personnel was administered to each student at the beginning of the course. These scores were also tested for homogeneity of variance, and that assumption was supported. All classes were given reading assignments, written assignments, and programming problems to solve. Some of the students used mark sense cards exclusively to prepare their programs; some used keypunch cards exclusively; while others used both. No attempt was made to control the type of cards used by the students.

At the end of the course, a multiple choice achievement test was administered to all students to test their understanding of elementary concepts of FORTRAN computer programming. The results were evaluated using analysis of covariance, with the set of ATPP scores as covariate. A test for homogeneity of regression was conducted, and that assumption was supported. The findings with respect to the hypotheses are presented below.

Hypothesis 1: The hypothesis concerning no significant differences in achievement between students in direct association with the computer at the three levels of frequency tested, one hour per term, one hour per week, and three hours per week, was rejected. Post-hoc analysis was conducted using the Tukey method to test all pairwise comparisons, and significant differences were found. Students in the group in direct association with the computer three hours per week scored significantly higher than each of the other two levels of frequency. The group in direct association only one hour per term scored significantly higher than the one hour per week group. This finding was contrary to anticipated results, and would seem to indicate one of two things: either (1)the treatments were not applied conscientiously, or, (2)other external factors were operating which influenced the treatment effects.

The questionnaire was checked to determine if the treatments were indeed perceived by the students as designed. One of the classes included eight students who reported they were in direct association with the computer three hours per week as opposed to the designated treatment of one hour per week. This particular class, however, had the highest adjusted mean score on the final of the three classes in that treatment group. The conclusion was made that, since only three per cent of the students reported deviations from frequency treatments, the treatments were applied in a manner which would not adversely affect the experiment. Hence, the question of extraneous factors affecting the treatments was checked, again referring to the questionnaire.

Item three of the questionnaire refers to the number of times required to process a program until a correct solution is obtained. Nearly 55% of the students reporting two attempts required to achieve successful results were in the treatment group H1, one hour per week in the computer laboratory. Of the 76 students in treatment group H1, 71% reported exclusive use of the keypunch machine to prepare program decks. This accounted for 60% of all students who reported exclusive use of the keypunch machine. Seventy-five of these students reported a waiting time of more than five minutes for a keypunch, including fifty who reported waiting up to twenty minutes. These 75 students accounted for the 47% of all students who reported waiting times for the keypunch.

Of the 112 students who attributed a "few" of their program errors to mistakes made in keypunching, 56 were in the H1 treatment level (50%). All but ten of the 76 students in treatment group H1 expressed a preference for more time in the computer laboratory, (Item 7 of the questionnaire). In summarizing these statistics, it would seem that a disproportionate number of students who experienced long waiting times for the keypunches, who reported keypunch errors, and who expressed a desire to be in another treatment group, were in treatment group H1. The conclusion seems justified that some effects of keypunching programs were detrimental to the performance of these students on the learning of FORTRAN concepts, hence on performance on the final exam.

Hypothesis 2: The hypothesis concerning no difference in achievement between students experiencing the three treatment levels of turn-around time was rejected. The group having to wait one week for the return of processed programs scored significantly lower on the final exam than either the group experiencing less than one day or the group experiencing about a one day delay. This appears to be supportive of Sackman's assertion that delay of feedback under the batch processing mode could have been a factor in his study, (Sackman, 1970). He did not investigate this factor, but proposed such a study be done.

It does, however, contradict the findings of some of the psychological studies reviewed which reported relatively small effects on learning due to delay of feedback and reinforcement schedules, (Fulton, 1969; Gay, 1972; Sassenrath, 1969).

An explanation for these findings was sought by referring to the student questionnaire. Each of the three classes under treatment level PW (one week turn-around time) reported on their perceptions of treatment F (frequency). There were no reported deviations from the design with respect to frequency among these three groups. Four members of the groups reported turn-around times other than those actually applied in the treatment. This was deemed too small a number to be detrimental to the actual treatments. Of the 198 students reporting two or more attempts needed to obtain a successful program, 54 were in the PW treatment group. Thirty-three reported using mark sense cards exclusively, while 14 used punch cards exclusively. Twenty members of this group expressed satisfaction with a one week turn-around time, forty would have preferred immediate turn-around time. None of these findings seemed to be deviating excessively from expected results.

The findings of this study support the assertion that delays in feedback (turn-around time) in a beginning course in computer programming have a detrimental effect on performance on an achievement test.

Hypothesis 3: The hypothesis concerning no difference in attitude toward the computer between students in direct association with the computer one hour per term, one hour per day, or one hour per week, was not supported.

The students in the H1 treatment group, one hour per week in the computer laboratory, expressed a significantly more positive attitude than those in the laboratory three hours per week. This could be due in part to the fact that students in the laboratory one hour per week had spent most of that time doing keypunching of program decks. As mentioned earlier, these same students expressed a high degree of frustration due to keypunching. Most of the students in the H0 group used mark sense cards. Thus, frustrations which they felt were attributed almost entirely to the computer. These students felt that programs returned as nonexecutable or incorrect were not accepted by the computer, rather than attribute their problems to mistakes which they may have made in marking. The students in the computer laboratory three hours per week used keypunch cards. However, because they had more time in the laboratory it was not necessary for them to rush to complete keypunching. Hence, they made fewer errors and experienced less frustration in keypunching. They, too, directed their feelings of frustration at the computer rather than the keypunch. The conclusion then is that students in the H1 group expressed a more positive attitude toward the computer because they directed their frustrations at the keypunch machines instead of the computer. There is no real data to support this conclusion, but it would be a question to investigate in further studies.

Hypothesis 4: The hypothesis concerning no difference in attitude toward the computer among students subjected to three different turn-around time delays of less than one day, one day, or one week, was supported.

This would seem to indicate that immediacy of feedback is not crucial to a positive attitude among students in a beginning course in computer programming.

Significant differences were found in achievement due to certain combinations of turn-around time and direct association. In particular, students who were in the computer laboratory three hours per week scored higher in achievement with the exception of those students who were also subjected to a one day turn-around time. Thus the one day turn-around time had a deleterious effect on the performance of students who were in the computer laboratory three hours per week.

Discussion of Conclusions

There are two general conclusions resulting from the findings of this experiment. The first conclusion is that student achievement in a course in computer programming is affected by the manner in which the school system arranges the operation of the course. It was evident from the findings that turn-around time and frequency of direct association affect student performance on an achievement test. Of the two treatments, turn-around time had the greater impact. Students experiencing the shortest delay in feedback scored highest on the final examination. Sackman expressed concern about this form of delay of feedback as a contributing factor in his study, (Sackman, 1970).

Secondly, there are factors other than turn-around time and frequency of direct association which also greatly affect student achievement and attitude as indicated by this study. Among these factors are the frustrations of waiting for a keypunch machine and of committing errors in typing. Those students who were in the three one hour per week frequency groups accounted for more than 50% of all students expressing frustrations with keypunching problems. In this study, students who were almost never in the laboratory used mark sense cards. Hence, they did not experience the problems arising from having to keypunch. Those students who were in the laboratory for three hours per week did their own keypunching. However, they had considerably more time in which to use the machines, and as a result, did not express concerns related to keypunching to as great a degree.

Implications for Further Study

This investigation indicated the effects on achievement and attitude toward the computer of two inherent characteristics of batch mode processing, namely, turn-around time and frequency of direct association with the computer. It was found that frequency of association with the computer had a significant effect on both attitude and achievement. Turn-around time had a significant effect on achievement, but not on attitude.

There was significant interaction between the treatments, time and frequency of association, with respect to achievement. Certain combinations of the two treatments had deleterious effects on student achievement. Further investigation of this outcome should be carried out in order to determine why this should be. Perhaps there is some quantitative way of expressing a relationship between attitude and achievement.

One of the findings in this study indicated that students in direct association with the computer three hours per week exhibited a significantly less positive attitude than students in direct association one hour per term or one hour per week. This finding was not expected and needs further investigation. The conclusion reached in this study is that there were some other factors operating which may have been more influential on student attitude than association with the computer. Among these factors are the effects of keypunching. Questions to be considered are:

(1) Does keypunching of program decks have an effect on student attitude in a course in computer programming?

(2) Do students who receive extensive training in the use of keypunch machines reflect better attitudes in a computer programming course than students who are not trained?

(3) Do students who use mark sense cards reflect a more positive attitude than students who use punched cards?

Such a study should attempt to control as rigidly as possible turn-around time and frequency of direct association with the computer. It should be carried out over a much shorter time span than this study. Alternative ways of preparing program decks would be:

(1) Have the decks punched by business students already trained to use the keypunch.

(2) Train the students to use the keypunch efficiently.

(3) Use mark sense cards exclusively, thereby, eliminating keypunching problems.

This study indicates that factors such as immediacy of feedback and frequency of association with the computer do affect student attitude and achievement. These factors can be controlled by the teacher of the course. Further study should be done to determine the most effective combination of these factors. Other time frames need to be investigated, such as daily association with the computer and immediate feedback.

Limitations Imposed on This Study

The following factors are considered to have imposed limits on the study.

(1) Lack of randomization of students in assignment to treatment groups. It was hoped that matching the groups on a Chi square test basis and using the analysis of covariance design would tend to alleviate effects of this.

(2) Limited control over the instruction given to the students. The number of students involved required several instructional groups and as a result several teachers. The teachers brought with them varying degrees of expertise in teaching computer programming and varying degrees of enthusiasm for the project. Although the teachers willingly agreed to participate in the study, some were more thorough than others in adhering to the treatments. For instance, on at least one occasion, one teacher whose class was not scheduled to attend the lab more than once per week permitted the students to go to the lab individually, on a more frequent basis. This situation was corrected rather early in the term, however.

(3) Mechanical problems due to computer down-time. On several occasions the computer was inoperative during the one period

> when a class was scheduled in the lab. This meant that the class missed its session with the computer working for one week. The class was in the laboratory with the keypunch machines, however, and the computer highly visible.

This investigation was planned to assess the treatment effects in as natural a setting as possible rather than in a laboratory one. For this reason, the effects discussed above would tend to enhance this aspect of the study.

APPENDIX A

COURSE OUTLINE

Computer Programming - Course Outline

I. Introduction

A. Computers in general

1. Discussion of Topic One of CWTC
2. Other computers; e.g., Programma 101, etc.

II. Elementary FORTRAN Concepts

A. Arithmetic Instructions

1. FORTRAN variables and symbols of operation, +,-,*,/,**
2. FORTRAN arithmetic statements
3. Order of operations
4. REAL and INTEGER variables and expressions

B. Output of numeric data

1. Simple WRITE and FORMAT statements

   Student instructed to use

   ```
     WRITE(5,2)X
   2 FORMAT(F16.4)
   ```

   or similar statements until further notice, for outputting REAL numerical quantities.
2. Simple QUOTE expressions, such as

   ```
     WRITE(5,3)X
   3 FORMAT(1X,'ANSWER IS',F16.4)
   ```

   Always begin any FORMAT statement containing QUOTE field, as above, with 1X.

C. Input

1. Assigning values, such as A = 5.0
   B = 10.0, etc.
2. READ statements and data cards
   a. One number per card, using F16.4
   b. More than one number per card, such as 5F16.4
      Student always instructed to punch decimal point on data card.

D. Branching

1. Flow charts
   a. Meaning of symbols, reading of charts
   b. Preparation of charts involving decision statements.
2. GO TO statement
3. IF statement
   a. Arithmetic IF; e.g. IF(X)10,20,30
   b. Logical IF; e.g. IF(X.EQ.0.0)GO TO 50
      Can be used on punch cards only.
   c. CONTINUE statement.

Suggested Programs and Assignments

Text: Communicating with the Computer
Referred to as CWTC

Lesson	Topic	Text Assignment	Program Assignment
1.	Introduction	Topic One	Average of Semester Grades, See Exercise Three, p.9
2.	Keypunching	Topic Two	Same, this time student punches and processes program.
3.	Arithmetic Instructions Outline, IIA, 1-3	Topics 3,4	Difference of Squares, Diff. of Cubes, assigning values to variables
4.	Output, Outline IIB1,2	Topic 9	Pythagorean Theorem-finding c in $c^2=a^2+b^2$, using C=(A**2+B**2)**.5, and labelling the answer.
5.	Input, Outline C2a,b	Topics 8,9	Average of grades, reading in grades, printing grades and labelling input and answer.
6.	Flow Charts	Topic 10	None assigned.
7.	Branching	Topic 11	Roots of a quadratic equation using IF statement to determine nature of roots.
8.	Branching	Topic 12	Same
9.	Iteractive Proc. Trailer card	Topic 13	Accumulating a sum, finding average; use trailer card to terminate program.
10.	DO loops INTEGER arith.	Topic 15 Topics 6,7	Printing squares or cubes of integers using DO loop to generate same
11.	DO loops	Same	Same

12.	Arrays, DIMENSION statement	None	Accumulating a sum, using subscripted variables.
13.	Same	None	Same
14.	Self-indexing READ/WRITE statements	None	Same, using self-indexing input/output statements
15.	Sorting	None	Find the largest/smallest number in a list.
16.	Class problem	None	Class problem.

APPENDIX B

ATTITUDE INSTRUMENT

____
Name and section (print)

Computer Attitude Instrument

This paper is designed to assess your feelings and attitudes toward the computer as it relates to various aspects of everyday life.

Instructions: The statements which follow reflect an attitude or belief about the computer. You are to indicate the extent to which you agree with the statement, or disagree with the statement, by placing X in the appropriate blank space. Do not skip any statements, and mark only one space for each statement.

	SD	D	NO	A	SA
Example 1. High school teachers are intelligent.	___	___	___	___	___

The letters above each blank represent:

SD - Strongly Disagree
D - Disagree
NO - No Opinion
A - Agree
SA - Strongly Agree

If you agree with the above statement, but do not strongly agree, you would place X in the blank under letter A. Your answer would appear as follows:

	SD	D	NO	A	SA
High school teachers are intelligent.	___	___	___	X	___
Example 2: Vice principals are unfair.	___	___	___	___	___

If you strongly disagree with this statement, you would place X in the blank under SD, as follows:

	SD	D	NO	A	SA
Vice principals are unfair.	X	___	___	___	___

If there are no questions, you may begin.

\- - - - - - - - - - - - - - - - - - -

____
Name and section-print

Computer Attitude Instrument

	SD	D	NO	A	SA
1. The computer is useful in learning mathematics.	___	___	___	___	___
2. In science and industry, the computer is no good.	___	___	___	___	___
3. The computer is useful with respect to society.	___	___	___	___	___
4. The computer is no good in education.	___	___	___	___	___
5. Use of the computer in science and industry is harmless.	___	___	___	___	___
6. The computer in society is worthless.	___	___	___	___	___
7. The computer in education is safe.	___	___	___	___	___
8. The computer in science and industry is safe.	___	___	___	___	___
9. The computer in society is harmless.	___	___	___	___	___
10. In science and industry, the computer is harmful.	___	___	___	___	___
11. The computer in society is dangerous.	___	___	___	___	___
12. In education, the computer is harmful.	___	___	___	___	___

Thank you for your cooperation.

APPENDIX C

ITEM ANALYSIS OF ATTITUDE INSTRUMENT

ITEM NO.	ITEM WEIGHT	CORRELATION WITH TOTAL	NUMBER GIVING EACH RESPONSE 0	1	2	3	4	5	6	7
1	1	0.3156	3	15	67	21	121	25		
2	1	0.4462	1	6	18	17	84	126		
3	1	0.4323	0	13	49	67	102	21		
4	1	0.4643	1	5	15	22	98	111		
5	1	0.3891	0	15	50	65	92	30		
6	1	0.4524	1	7	27	53	93	71		
7	1	0.5568	0	15	28	65	116	28		
8	1	0.5830	2	15	39	64	104	28		
9	1	0.5649	0	11	57	76	92	16		
10	1	0.7029	0	3	22	66	103	58		
11	1	0.6315	0	6	29	55	103	59		
12	1	0.6417	1	5	15	49	110	72		

N = 252 MEAN = 43.273 STDV = 6.384

RELIABILITY UNCORRECTED ODD-EVEN = 0.6471 CORRECTED ODD-EVEN = 0.7857

APPENDIX D

FINAL EXAMINATION

All answers are to be placed on the special answer sheet provided. Use only a dark lead pencil. Fill in the space completely, do not make any stray marks or mark outside the bubble. There is only one correct answer per question.

Part I. Multiple choice.

1. Which of the following statements is true?

   a. The computer can correct a wrong gas meter reading.
   b. The computer can think original thoughts.
   c, The computer can do arithmetic.
   d. The computer cannot store information.
   e. None of the above is a true statement.

2. Which of the following FORTRAN statements is equal to $y = 3x^2$?

   a. Y = 3.0*X*2   b. Y = (3.0*X)*2   c. Y = 2.0*(A**2+B**2)
   c. Y = 3.0*X**2   e. Y = (3.0*X)**2

3. Which of the following is not a correct FORTRAN statement?

   a. X = 2.0*(A**2+B**2)**0.5   b. X = 2.0*(A**2+B**2)
   c. X = 2.0*((A**2)+(B**2))**2   d. X = 2.0*(A+(-B))
   e. X = 2.0*(A**2+B**2))**3

4. In the expression X+A*B/C, the operation performed first is

   a. Addition   b. Multiplication   c. Division
   d. Exponentiation   e. Subtraction

5. In the expression (R+S)*A**2-C/3.0, the first operation performed is

   a. Addition   b. Multiplication   c, Division
   d. Exponentiation   e. Subtraction

6. For the statement READ(8,1)X,Y an appropriate FORMAT statement is

   a. 8 FORMAT(2F16.4)   b. 8 FORMAT(F16.4)   c. 1 FORMAT(F16.4)
   d. 1 FORMATF16.4   e. None of these choices if correct

7. Which of the following is a correct pair of input statements?

a.
```
  READ(8,3)R
1 FORMAT(F10.2)
```
b.
```
  READ R
1 FORMAT(F10.2)
```
c.
```
  READ(8,3)R
3 FORMATF10.2
```
d.
```
  READ(8,3)R
8 FORMAT(F10.2)
```
e.
```
  READ(8,3)R
3 FORMAT(F10.2)
```

8. Which of the following pair of input statements would be used to read in the numbers 3.142, 1.976, 1.414, and 1.001, with all four numbers appearing on the same card, punched starting in column 1, with no spaces?

a.
```
  READ(8,1)ABCD
1 FORMAT(F4.0)
```
b.
```
  READ(8,1)A,B,C,D
5 FORMAT(4F4.3)
```
c.
```
  READ(8,1)A,B,C,D
8 FORMAT(4F5.3)
```
d.
```
  READ(8,1)A,B,C,D
1 FORMAT(4F5.3)
```
e.
```
  READ(8,1)A,B,C,D
1 FORMAT(F5.3)
```

9. Which of the following pairs of output statements would cause the computer to print the message ANSWER IS spaced as shown?

a.
```
  WRITE(8,2)
2 FORMAT(1X,'ANSWER IS')
```
b.
```
  WRITE(5,2)
2 FORMAT(1X,'ANSWER IS')
```
c.
```
  WRITE(5,2)ANSWER IS
2 FORMAT(F16.4)
```
d.
```
  WRITE(5,2)
2 FORMAT('ANSWER IS')
```
e.
```
  WRITE(5,2)
2 FORMAT('ANSWER IS')
```

10. Given the program segment shown at the right, what would be the value of R after execution?

```
H=4.0
B=5.0
X=(B-H)*(B+H)
Y=B**2-H**2
R=X-Y
R=R+1.0
```

a. -18.0 b. 19.0 c. 9.0
d. 1.0 e. 0.0

11. Which of the following is a correct IF statement?

a. IF(X)10,20,30 b. IF X 10,20,30 c. IF(X,10,20,30
d. IF X (10,20,30) e. IF(X)(10,20,30)

12. Consider the statement IF(K-100)16,17,18 . What is the number of the next instruction executed by the computer given K=50?

a. 16 b. 17 c. 18 d. 100 e. 50

13. Select the IF statement which would branch to statement 18 if P is negative.

a. IF P 18,19,20 b. IF(P)20,19,18 c. IF(P-0.0)20,19,18
d. IF(P,18,19,20) e. IF(P)18,19,20

In answering questions 14 and 15, refer to the program shown at the right.

```
      READ(8,100)A
      Z = 5000.0
      IF(A-5.0)30,16,8
   30 WRITE(5,100)A
      CALL EXIT
   16 WRITE(5,100)A
      CALL EXIT
    8 XRAY = A**2
      WRITE(5,100)XRAY
      CALL EXIT
  100 FORMAT(F16.4)
```

14. If A = 9.0, what is the number of the next statement executed after the IF statement?

a. 5 b. 300 c. 16 d. 8 e. 100

15. What would be printed if A = 5.0?

a. 5000.000 b. 30 c. 5.0000
d. 25.0000 e. 8

16. Which of the following statements would cause the computer to branch to statement 50? unconditionally?

a. X = 50.0 b. BRANCH TO 50 c. IF(A)50,40,50 d. A = 50
e. GO TO 50

In answering questions 17 and 18, refer to the program at the right.

```
      READ(8,1)X,Y
      IF(Y)10,20,10
   10 A = X/Y
      GO TO 30
   20 A = Y/X
   30 WRITE (5,1)A
    1 FORMAT(F16.2)
      CALL EXIT
```

17. What would be printed if X = 10.0 and Y = 5.0?

a. 10.5 b. 2.00 c. 0.00
d. 2 e. 1/2

18. What would be printed if S = 3.0 and Y = 0.0?

a. 2.00 b. 0.500 c. 0.00 d. 2 e. 1/2

19. Which pair of statements would cause the computer to read in the number 57.03 and assign it to SPEED using statement number 1?

```
a.    READ(8,4)SPEED       b.    READ(8,1)SPEED       c.    READ(1,8)SPEED
    1 FORMAT(F5.2)             8 FORMAT(F5.2)             8 FORMAT(F4.5)

d.    READ(8,1)SPEED       e.    READ(5,8)SPEED
    1 FORMAT(F5.2)             1 FORMAT(F5.2)
```

20. Select the pair of statements below which would cause the computer to print the message DON'T GIVE UP THE SHIP in its entirety.

```
a.    WRITE(5,3)
    3 FORMAT(1X,DON'T GIVE UP THE SHIP)

b.    WRITE(5,3)
    3 FORMAT(1X'DON'T GIVE UP THE SHIP')

c.    WRITE(5,3)DON'T GIVE UP THE SHIP
    3 FORMAT(F16.4)

d.    WRITE(8,3)
    3 FORMAT(1X,'DON'T GIVE UP THE SHIP')
```

e. None is correct

21. The correct order of control cards is

```
a. // JOB              b. //JOB              c. // JOB
   // FOR                 // XEQ                // FOR
      ...                    ...                    ...
   // XEQ                  CALL EXIT                CALL EXIT
      CALL EXIT              END                      END
      END                                        // XEQ

d. // JOB              e. None is correct
      CALL EXIT
      END
```

22. Which of the following is a correct FORTRAN statement number?

a. 555555 b. -555555 c. 55555.5 d. 555 e. 5,555

23. If no parentheses appear in an arithmetic expression the first operation performed is

a. Addition b. Subtraction c. Multiplication d. Division
e. Exponentiation

24. Which of the following FORMAT codes would print
1976 1066 1492 1884 as shown?

a. 6I4 b. 4I7 c. 16I4 d. I16 e. 4I5

25. If K = 4/6, what is the value of K according to I4?

a. 1.5 b. 0 c. .66 d. 1.555... e. .6666

Part II

26. Write a complete computer program which will

- read in 5 numbers punched one per card, including decimal points,
- assign them to the variables A, B, C, D, and E
- find the average of the 5 values,
- print the answer following the message ANSWER IS

Assume the numbers will be given to you later. Omit control statements.

APPENDIX E

VALIDATION QUESTIONNAIRE

Validation Questionnaire

1. Please indicate the extent to which you feel this test measures student understanding of elementary concepts in FORTRAN programming.

   To a large extent ______

   To some extent ______

   Very little at all ______

   Not at all ______

2. Listed below are elementary concepts of FORTRAN programming. Indicate by placing a check in the appropriate blank how important you think it is that each concept appear in a test such as this.

	Very	Some-what	Not very	Not at all
Control statements	____	____	____	____
FORTRAN variables	____	____	____	____
Comment statements	____	____	____	____
FORTRAN order of operations	____	____	____	____
INTEGER mode and REAL mode	____	____	____	____
Input statements	____	____	____	____
Output statements	____	____	____	____
FORMAT statements	____	____	____	____
Statement numbers	____	____	____	____
Flow charting	____	____	____	____
IF statements	____	____	____	____
GO TO statements	____	____	____	____
DO loops	____	____	____	____

3. Refer to the test under consideration. Indicate the extent to which you feel each concept is tested.

	Very	Some-what	Not very	Not at all
Control statements	____	____	____	____
FORTRAN variables	____	____	____	____
Comment statements	____	____	____	____
FORTRAN order of operations	____	____	____	____
INTEGER mode and REAL mode	____	____	____	____
Input statements	____	____	____	____
Output statements	____	____	____	____
FORMAT statements	____	____	____	____
Statement numbers	____	____	____	____
Flow charting	____	____	____	____
IF statements	____	____	____	____
GO TO statements	____	____	____	____
DO loops	____	____	____	____

APPENDIX F

VALIDATION RESULTS

Validation Questionnaire

Number in blank represents score rating.

1. Please indicate the extent to which you feel this test measures student understanding of elementary concepts in FORTRAN programming.

To a large extent	4
To some extent	3
Very little at all	2
Not at all	1

2. Listed below are elementary concepts of FORTRAN programming. Indicate by placing a check in the appropriate blank how important you think it is that each concept appear in a test such as this.

	Very	Somewhat	Not very	Not at all
Control statements	4	3	2	1
FORTRAN variables				
Comment statements				
FORTRAN order of operations				
INTEGER mode and REAL mode				
Input statements				
Output statements				
FORMAT statements				
Statement numbers				
Flow charting				
IF statements				
GO TO statements				
DO loops				

Average scored value = 3.46

3. Refer to the test under consideration. Indicate the extent to which you feel each concept is tested.

	Very	Somewhat	Not very	Not at all
Control statements	4	3	2	1
FORTRAN variables				
Comment statements				
FORTRAN order of operations				
INTEGER mode and REAL mode				
Input statements				
Output statements				
FORMAT statements				
Statement numbers				
Flow charting				
IF statements				
GO TO statements				
DO loops				

Average scored value = 2.93

Note: Scores less than 2.5 indicate disagreement.

Validation Questionnaire

1. Please indicate the extent to which you feel this test measures student understanding of elementary concepts in FORTRAN programming.

To a large extent	9
To some extent	
Very little at all	
Not at all	

2. Listed below are elementary concepts of FORTRAN programming. Indicate by placing a check in the appropriate blank how important you think it is that each concept appear in a test such as this.

	Very	Somewhat	Not very	Not at all
Control statements	9			
FORTRAN variables	9			
Comment statements	9			
FORTRAN order of operations	9			
INTEGER mode and REAL mode	3	6		
Input statements	6	3		
Output statements	6	2	1	
FORMAT statements	9			
Statement numbers	9			
Flow charting	2	6	1	
IF statements	3	6		
GO TO statements	3	5	1	
DO loops	3	4	2	

3. Refer to the test under consideration. Indicate the extent to which you feel each concept is tested.

	Very	Somewhat	Not very	Not at all
Control statements	3	0	6	
FORTRAN variables	9			
Comment statements	3	3	3	
FORTRAN order of operations	9			
INTEGER mode and REAL mode	3	6		
Input statements	3	6		
Output statements	3	6		
FORMAT statements	7	2		
Statement numbers	3	6		
Flow charting			2	7
IF statements	7	2		
GO TO statements			3	6
DO loops				9

List of judges requested to validate final examination

Judge	Position
F. Gilbert French	Head, Baltimore City College Baltimore Public School System
Zeney Jacobs	Assistant Principal School 403, Baltimore Public Schools
Lee Jones	Instructor of Computer Programming Edmondson High School Baltimore City Public Schools
Anthony Konstant	Instructor of Computer Programming Baltimore Polytechnic Institute Baltimore Public Schools
M. Carroll Paugh	Instructor of Computer Programming School 403, Baltimore Public Schools
Linda Rosenberg	Instructor of Computer Programming Patterson High School Baltimore Public Schools
Jacob Schuchman	Mathematics Specialist Baltimore City Public Schools
Charles Sullivan	Instructor of Computer Programming Baltimore Polytechnic Institute Baltimore Public Schools
Irvin Yaffe	Instructor of Computer Programming School 403 Baltimore City Public Schools

APPENDIX G

STUDENT QUESTIONNAIRE

AND

SUMMARY OF RESULTS

Computer Programming Course - Questionnaire

Your response to each of the following questions is most important. Carefully read each question, then answer by checking in the appropriate blank, or writing a brief answer, as indicated.

1. Estimate the number of class periods per week you were in the computer lab.

   92 2 periods/week   76 1 period/week   84 No periods/week

2. Estimate the amount of time delay between the time you submitted for processing a deck of computer cards and the time your received the printout.

   80 About one week   87 About one day   85 Less than one day

3. Estimate, on the average, the number of times required to process a program until a correct printout was received.

   54 One time   160 Two times   38 More than two times

4. How many programs did you process using punch cards?

   84 None   48 More than half, but not all

   30 At least one, but less than half   90 All

   If you used the keypunch for more than one program, answer questions a, b, and c below. Otherwise, skip to question 5.

   a. Estimate the amount of time spent waiting for a keypunch.

      11 5 minutes or less   76 Between 10 and 20 minutes

      25 Between 5 and 10 minutes   47 More than 20 minutes

   b. Estimate the number of program errors due to mistakes in keypunching.

      14 None   112 A few   28 Most

   c. In your opinion, the greatest advantage of keypunching is

      31 Convenience   76 Speed   37 Accuracy   12 Handling

5. How many programs did you process using mark sense cards?

112 None

19 More than half, but not all

28 At least one, but less than half

84 All

If you used mark sense cards for more than one program, answer questions a, b, and c. Otherwise, skip to question 6.

a. Estimate the amount of time spent marking cards

7 5 minutes or less

82 Between 10 and 20 minutes

34 Between 5 and 10 minutes

15 More than 20 minutes

b. Estimate the number of program errors due to mistakes made in marking.

12 None   95 A few   25 Most

c. In your opinion, the greatest advantage in mark sense cards is

64 Convenience   14 Speed   23 Accuracy   14 Handling

6. Did you use both punch cards and mark sense cards for more than one program?

43 Yes   167 No   If so, which did you prefer?   27 Punch

16 Mark sense

7. Given your choice of how frequently your class would go to the computer lab each week, which of the following would you choose?

204 3 periods/week   23 1 period/week   14 No periods/week

8. Given your choice of time delay between submitting a card deck and receiving the printout, which of the following would you choose?

29 About one week   64 About one day   128 Less than one day

9. Assume grades for this course were assigned as follows: A-excellent, B-good, C-fair, D-poor. Which grade would you assign yourself? Be honest, please.

______ Grade

10. State briefly what you liked best about this course. ______________

11. State briefly what you disliked most about this course. ___________

Thank you for your cooperation.

APPENDIX H

RAW SCORES ATTITUDE INSTRUMENT

STUD. NO.	ITEM NUMBER 1	2	3	4	5	6	7	8	9	10	11	12
10101	2	5	4	2	1	5	1	1	4	4	3	3
10102	3	5	4	5	5	5	4	4	4	5	5	5
10103	2	4	4	4	4	4	2	2	2	2	2	4
10104	2	2	1	4	2	3	4	2	3	3	5	4
10105	4	5	4	4	2	4	4	4	4	4	4	4
10106	4	5	4	2	4	2	3	3	2	4	4	5
10107	2	4	4	4	2	4	4	4	2	4	4	4
10108	3	5	4	5	3	5	3	3	3	5	5	5
10109	4	5	4	4	2	4	4	4	2	4	2	4
10110	4	4	2	4	4	4	4	4	4	4	4	4
10112	4	4	3	5	2	2	5	2	1	4	4	5
10113	4	5	3	5	4	5	4	4	4	4	5	5
10114	5	4	1	5	4	4	1	1	4	4	4	4
10115	2	4	4	4	4	5	5	5	4	5	4	4
10116	2	4	2	4	4	4	2	2	2	4	4	4
10117	4	5	3	4	4	5	4	4	2	4	4	5
10119	4	5	4	5	5	3	4	4	3	5	3	5
10121	4	5	4	5	4	5	5	5	4	5	5	5
10122	5	5	5	5	5	5	5	5	5	5	5	5
10123	2	5	2	4	4	2	4	2	4	2	2	3
10124	4	5	4	5	5	5	3	4	4	4	5	4
10126	3	4	4	4	5	3	2	5	3	5	3	2
10127	4	5	4	5	3	5	2	4	2	4	2	3
10128	2	2	2	4	4	4	1	4	4	4	4	1
10129	4	4	2	4	2	4	4	4	2	4	4	4
10130	4	5	4	5	2	4	1	2	2	4	4	4
10131	2	5	4	5	2	5	4	3	2	4	3	5
10134	2	4	4	4	4	5	4	4	4	5	5	5

STUD.	ITEM NUMBER											
NO.	1	2	3	4	5	6	7	8	9	10	11	12
10201	4	4	4	4	4	4	4	4	4	4	4	4
10203	1	4	3	2	3	4	4	3	3	3	3	3
10204	3	5	4	5	4	3	3	4	3	3	3	4
10205	2	4	2	5	4	3	5	4	2	4	4	5
10207	2	4	2	5	2	3	3	2	3	4	4	3
10209	4	4	3	2	1	4	4	2	2	4	2	4
10210	3	5	4	4	4	2	3	2	3	4	4	3
10211	4	1	3	4	3	4	4	4	3	3	4	4
10212	2	4	2	4	4	3	1	1	2	2	3	2
10213	5	5	4	5	3	5	5	4	3	4	3	5
10214	3	4	4	2	3	4	3	3	3	4	3	2
10215	2	4	3	5	4	4	5	4	4	4	4	4
10216	4	4	3	5	3	4	5	4	3	2	3	4
10218	4	5	3	5	4	5	5	5	5	5	5	5
10219	2	4	3	3	3	3	4	4	3	3	3	3
10221	2	4	4	4	4	3	2	3	3	3	2	2
10222	4	4	4	4	3	4	4	3	4	4	4	5
10223	4	4	4	4	3	4	4	3	4	4	4	5
10224	4	5	5	5	4	5	4	4	4	5	5	5
10225	3	4	4	4	4	4	3	3	3	4	3	3
10226	2	3	4	3	3	3	3	2	4	3	3	4
10227	2	4	3	4	3	4	2	2	2	4	4	4
10228	3	4	4	4	3	3	3	4	4	4	4	4
10229	2	4	4	5	3	5	4	4	4	3	3	4
10230	4	4	4	4	3	4	4	3	4	4	4	5
10231	2	3	3	3	5	5	3	3	2	3	3	3
10232	4	5	4	4	4	3	3	3	3	4	5	4
10233	4	4	4	4	4	4	4	4	4	4	4	4

STUD. NO.	1	2	3	4	5	6	7	8	9	10	11	12
	ITEM NUMBER											
10301	2	3	3	4	3	3	2	3	3	3	3	2
10303	4	5	4	5	2	4	2	2	2	2	2	4
10304	3	5	1	5	4	4	2	0	3	5	2	3
10305	3	2	2	3	4	3	4	3	3	3	2	3
10306	1	5	5	5	5	5	5	5	5	5	5	5
10307	3	4	3	5	4	5	1	3	5	4	5	5
10308	4	5	5	5	3	4	3	4	2	3	2	3
10309	2	4	3	5	2	2	3	4	1	4	1	5
10310	2	5	3	4	2	3	4	5	3	5	4	4
10311	2	5	4	5	3	5	3	5	3	4	5	5
10312	4	4	3	5	3	3	3	3	3	3	3	4
10313	4	3	2	4	3	2	3	3	2	2	4	4
10314	3	5	3	5	4	2	4	4	4	4	4	4
10316	4	4	3	5	3	4	2	2	4	3	4	2
10318	4	5	4	4	2	4	4	4	3	4	3	4
10319	2	5	2	4	4	3	4	4	4	4	3	4
10320	4	4	1	4	2	3	3	4	1	2	2	4
10321	1	2	4	3	2	1	3	2	2	2	2	2
10322	2	5	3	3	2	5	3	3	3	3	3	3
10323	4	4	3	5	4	5	4	4	2	4	5	5
10324	2	5	3	5	3	3	4	3	3	3	3	5
10325	2	4	4	2	3	3	2	4	3	4	4	3
10327	5	5	5	1	2	2	4	4	4	5	5	5
10328	5	5	5	5	3	3	5	3	4	5	5	3
10329	2	4	2	0	4	4	2	2	4	4	4	4
10330	2	1	3	5	1	3	1	2	3	2	4	3
10331	2	4	4	2	3	4	3	2	2	2	4	2
10332	1	5	5	5	5	5	5	5	5	5	5	5

STUD. NO.	ITEM NUMBER 1	2	3	4	5	6	7	8	9	10	11	12
10401	2	5	2	5	4	5	4	4	4	5	4	4
10402	4	5	2	5	3	5	5	3	2	4	5	5
10403	4	4	3	5	2	4	4	2	2	4	3	4
10404	5	4	1	4	4	2	4	4	1	3	2	3
10405	4	4	2	5	4	4	4	4	4	4	4	4
10406	5	4	4	5	2	2	4	4	4	4	4	4
10407	1	4	4	5	4	4	4	4	5	5	4	5
10411	4	3	4	5	4	4	4	3	3	4	4	4
10413	4	4	3	5	2	4	4	3	2	2	4	2
10414	4	3	3	3	3	3	4	4	4	3	3	3
10415	2	4	5	5	4	5	2	3	5	4	3	2
10416	4	4	3	4	2	2	2	3	2	2	4	3
10418	4	5	2	5	4	5	5	5	5	4	5	5
10419	4	4	4	4	3	5	4	3	3	3	5	4
10420	4	4	5	5	4	5	5	5	5	5	5	5
10421	4	4	4	5	3	5	4	3	4	4	4	5
10422	4	5	4	5	3	4	3	3	4	3	3	4
10424	2	4	4	4	5	4	4	2	3	3	3	4
10425	2	5	2	2	2	1	1	1	1	2	2	3
10426	2	4	2	4	3	4	2	3	4	3	4	4
10427	5	5	3	2	4	5	1	4	4	5	5	3
10428	5	3	3	5	3	4	4	4	4	5	5	4
10429	5	5	3	5	1	5	5	5	3	5	5	5
10430	4	4	2	5	4	3	4	4	4	5	4	5
10431	4	5	4	5	4	4	4	4	4	4	4	4
10433	5	5	4	5	2	4	4	3	2	3	4	4
10434	4	5	4	5	4	2	2	1	3	4	4	4
10435	5	5	2	5	2	3	4	5	2	3	1	2

STUD. NO.	ITEM NUMBER 1	2	3	4	5	6	7	8	9	10	11	12
10501	4	4	3	4	2	3	3	4	3	4	3	4
10502	1	5	1	5	5	1	5	1	5	5	5	5
10503	2	4	1	5	3	3	4	3	3	3	3	4
10504	4	5	4	5	4	5	1	1	4	5	4	4
10505	4	5	3	4	5	4	4	4	4	4	4	4
10506	5	5	4	4	4	3	4	4	4	4	4	4
10507	4	1	3	5	4	2	3	4	3	2	3	2
10508	4	4	3	4	2	4	3	3	3	5	4	4
10509	4	4	4	5	1	4	4	4	4	4	4	4
10510	4	4	3	4	3	4	3	3	4	4	4	4
10511	4	5	4	5	4	4	3	4	4	5	4	3
10512	5	3	4	5	3	4	4	3	4	3	5	5
10513	2	5	4	4	4	5	4	5	4	4	2	4
10514	4	5	2	5	5	5	4	4	4	5	5	5
10515	4	2	1	4	3	5	1	1	1	1	1	1
10516	1	4	4	3	4	2	3	1	4	3	2	5
10517	4	3	2	4	4	2	3	3	4	2	3	3
10518	4	5	4	5	4	5	5	4	2	4	4	4
10519	4	3	4	5	4	3	5	5	4	5	4	5
10520	4	4	2	4	2	4	3	2	2	3	3	3
10521	4	2	5	5	2	5	4	5	1	4	3	5
10522	3	5	3	4	4	3	3	4	3	4	3	4
10523	4	5	3	4	4	0	4	4	3	5	5	5
10525	1	5	4	5	4	4	5	5	4	4	4	4
10526	4	5	4	5	4	3	2	2	3	3	4	4
10527	4	5	5	4	1	5	4	2	2	3	3	4
10528	4	0	1	2	4	4	4	4	4	4	2	3
10530	4	4	4	5	5	1	5	5	5	4	5	5

STUD.	ITEM NUMBER											
NO.	1	2	3	4	5	6	7	8	9	10	11	12
10601	5	4	4	5	4	5	4	4	4	4	5	5
10602	5	5	4	5	2	5	2	3	4	5	5	5
10603	4	5	4	4	4	4	4	4	4	4	2	3
10604	2	5	4	4	4	4	2	4	3	4	4	4
10605	4	5	4	4	3	5	3	3	3	2	2	4
10606	2	1	4	4	4	4	4	4	4	4	5	4
10607	4	5	4	4	4	4	4	4	2	4	4	4
10608	2	5	2	4	5	3	3	3	4	3	3	5
10609	2	5	5	3	3	5	2	5	3	5	5	3
10610	4	5	1	5	4	5	4	4	4	5	5	5
10611	2	5	3	5	3	3	3	2	2	2	4	4
10612	4	5	4	5	1	5	1	1	1	5	5	5
10614	2	4	3	4	4	4	3	3	3	4	4	4
10615	5	5	3	5	4	4	5	5	3	5	5	5
10617	2	5	4	3	3	5	5	5	5	5	5	4
10618	4	4	3	4	2	4	2	2	3	3	4	4
10619	5	5	1	4	2	2	3	2	2	3	2	5
10620	2	4	4	3	5	4	4	4	4	4	5	5
10621	4	5	3	5	1	5	3	4	4	2	5	5
10622	4	4	3	1	5	1	3	3	2	3	3	1
10623	2	5	5	4	2	5	3	1	2	3	4	3
10624	3	5	2	4	4	4	1	1	4	4	5	5
10625	3	5	4	2	3	4	4	3	3	5	3	5
10626	4	4	4	5	4	3	3	3	3	4	4	3
10627	3	1	3	5	2	1	3	3	3	3	4	2
10629	4	5	4	4	3	5	4	3	3	5	4	5
10631	4	5	4	4	4	4	4	4	4	5	4	4
10632	4	5	4	1	1	5	3	2	3	3	5	3

STUD. NO.	1	2	3	4	5	6	7	8	9	10	11	12
	ITEM NUMBER											
10701	2	5	3	4	3	4	3	4	3	4	4	4
10702	2	4	3	4	2	4	4	4	4	4	4	4
10703	4	5	5	4	5	4	5	4	5	3	1	5
10704	4	5	4	4	5	5	4	5	4	5	5	4
10705	4	5	2	4	1	2	4	2	2	4	3	4
10706	4	2	3	4	4	4	4	4	4	4	4	5
10707	4	5	4	5	3	5	4	4	3	3	3	4
10708	4	2	2	4	4	3	4	4	3	4	3	4
10709	2	5	5	5	2	5	4	4	2	4	4	5
10710	1	5	2	5	4	2	2	4	1	5	3	2
10711	4	4	4	5	2	4	4	4	2	2	2	0
10712	4	4	4	5	3	5	3	3	3	3	4	4
10713	5	5	5	5	3	5	5	5	3	5	5	5
10715	4	4	4	4	2	4	4	5	2	4	2	4
10716	4	5	2	5	4	5	3	4	4	4	4	4
10717	2	5	5	4	5	5	3	4	5	5	5	3
10718	4	5	3	4	5	2	4	4	4	5	5	5
10719	4	4	4	4	2	1	4	2	3	4	4	4
10720	5	5	3	4	2	3	5	1	2	1	1	1
10721	1	2	2	2	4	4	4	4	4	4	4	4
10722	2	5	4	4	3	3	4	4	3	5	5	5
10723	3	1	3	4	3	4	3	3	2	3	4	3
10724	4	5	3	4	4	4	3	3	3	3	3	3
10725	2	4	3	4	1	3	3	3	2	3	3	4
10726	4	2	2	5	4	2	3	3	3	5	3	4
10727	4	4	4	5	3	3	4	4	4	4	4	4
10728	1	5	3	4	3	4	4	4	1	3	4	4
10729	4	4	1	4	3	3	4	4	4	4	3	4

STUD. NO.	ITEM NUMBER 1	2	3	4	5	6	7	8	9	10	11	12
10801	4	4	2	4	2	4	4	4	2	4	4	4
10802	4	5	4	5	2	4	1	2	2	4	4	4
10803	2	5	4	5	2	5	4	3	2	4	3	5
10804	4	4	4	5	4	5	3	3	4	5	5	5
10805	2	4	4	4	4	5	4	4	4	5	5	5
10806	4	5	2	4	4	4	4	4	4	4	4	4
10807	3	5	2	4	5	3	4	4	3	4	4	4
10808	2	4	4	5	3	5	4	4	4	3	3	4
10809	0	5	2	4	4	5	4	4	4	4	4	4
10810	4	5	4	4	3	4	4	5	3	5	4	4
10811	4	2	2	5	5	4	4	4	5	5	4	4
10812	2	3	3	3	5	5	3	3	2	3	3	3
10813	2	5	5	4	3	4	3	3	3	5	4	4
10814	2	3	3	3	5	5	3	3	2	3	3	3
10815	2	5	4	5	3	4	3	4	3	4	5	5
10816	4	5	4	4	4	3	3	3	3	4	5	4
10817	4	5	4	1	1	5	3	2	3	3	5	3
10818	5	5	3	5	1	5	5	5	3	5	5	5
10819	4	4	2	5	4	3	4	4	4	5	4	5
10820	4	5	4	5	4	4	4	4	4	4	4	4
10821	4	5	2	5	5	4	3	2	3	4	4	4
10822	5	5	4	5	2	4	4	3	2	3	4	4
10823	4	5	4	5	4	2	2	1	3	4	4	4
10824	5	5	2	5	2	3	4	5	2	3	1	1
10826	4	5	4	4	4	4	4	4	4	5	4	4
10827	4	5	3	5	3	3	4	4	3	4	3	4
10829	4	5	2	5	5	4	2	2	3	4	4	4
10830	2	4	5	4	2	4	3	3	2	3	4	3

STUD. NO.	ITEM NUMBER 1	2	3	4	5	6	7	8	9	10	11	12
10901	3	2	2	3	1	2	2	2	2	3	2	3
10902	4	5	3	5	5	3	4	4	4	4	5	5
10903	3	5	4	4	4	5	4	4	4	4	4	4
10904	1	3	4	5	3	4	4	3	4	3	4	5
10905	2	2	2	2	2	3	3	4	3	3	3	3
10906	4	5	4	5	3	3	4	4	4	5	5	5
10907	4	5	3	3	4	3	4	4	3	3	2	3
10908	4	4	2	1	4	5	4	4	4	5	5	5
10909	5	5	4	5	2	5	5	2	2	4	4	4
10910	4	3	3	5	4	2	4	4	4	3	3	3
10911	2	5	2	3	2	2	3	3	4	2	4	5
10912	0	2	3	4	3	4	4	4	4	3	4	4
10913	5	4	1	3	3	4	2	2	1	1	2	3
10914	4	4	4	4	4	2	2	2	4	4	4	4
10915	1	3	3	3	3	3	3	3	3	3	3	3
10916	2	3	2	4	3	3	4	3	4	3	3	3
10917	4	5	5	5	3	5	4	5	3	3	5	5
10918	1	5	4	4	4	4	4	2	3	4	5	4
10920	4	5	3	5	5	3	4	0	4	5	5	4
10922	4	5	2	2	4	4	4	1	2	3	2	3
10923	3	2	2	3	1	2	2	2	2	3	2	3
10924	2	5	2	3	2	2	3	3	4	2	4	5
10925	0	2	3	4	3	4	4	4	4	3	4	4
10926	4	5	3	3	4	3	4	4	3	3	2	3
10927	4	2	2	5	5	4	4	4	5	5	4	4
10928	2	3	3	3	5	5	3	3	2	3	3	3
10930	2	4	5	4	2	4	3	3	2	3	4	3
10931	1	2	3	2	3	4	1	2	2	2	2	2

APPENDIX I

RAW SCORES MANOVA

STUD. NO.	V A R I A B L E ARITH	ATPP	FINAL	ATTIT
10101	10.6	50.0	13.0	35.0
10102	10.8	57.0	19.0	54.0
10103	11.0	58.0	18.0	36.0
10104	10.2	57.0	16.0	35.0
10105	9.5	48.0	16.0	47.0
10106	9.8	48.0	17.0	42.0
10107	9.7	35.0	16.0	42.0
10108	9.8	38.0	14.0	49.0
10109	10.8	42.0	18.0	43.0
10110	10.4	51.0	16.0	46.0
10112	10.8	44.0	16.0	41.0
10113	10.3	43.0	19.0	52.0
10114	10.1	53.0	16.0	41.0
10115	11.5	41.0	13.0	50.0
10116	12.6	74.0	20.0	38.0
10117	11.0	58.0	14.0	48.0
10119	12.0	75.0	19.0	50.0
10121	12.6	60.0	17.0	56.0
10122	11.5	44.0	19.0	60.0
10123	11.7	45.0	16.0	36.0
10124	10.6	46.0	17.0	52.0
10126	11.3	66.0	16.0	43.0
10127	11.5	41.0	18.0	43.0
10128	10.4	46.0	18.0	36.0
10129	10.6	46.0	15.0	42.0
10130	10.6	50.0	17.0	41.0
10131	8.8	33.0	16.0	44.0
10134	11.0	57.0	17.0	50.0

N = 28

STUD. NO.	VARIABLE ARITH	ATPP	FINAL	ATTIT
10201	10.7	46.0	12.0	48.0
10203	9.2	38.0	15.0	36.0
10204	10.8	59.0	19.0	44.0
10205	9.3	31.0	13.0	44.0
10207	8.7	37.0	14.0	37.0
10209	9.3	30.0	14.0	36.0
10210	9.4	34.0	16.0	41.0
10211	10.0	48.0	18.0	41.0
10212	9.5	36.0	14.0	30.0
10213	12.0	61.0	22.0	51.0
10214	10.3	48.0	20.0	38.0
10215	11.3	35.0	17.0	47.0
10116	10.3	48.0	16.0	44.0
10218	10.6	32.0	14.0	56.0
10219	10.3	46.0	19.0	38.0
10221	8.4	48.0	8.0	36.0
10222	10.1	45.0	14.0	47.0
10223	11.4	53.0	19.0	47.0
10224	10.6	46.0	17.0	55.0
10225	8.8	31.0	16.0	42.0
10226	11.7	42.0	18.0	37.0
10227	10.6	40.0	21.0	38.0
10228	11.4	57.0	21.0	44.0
10229	10.9	50.0	15.0	45.0
10230	9.7	37.0	11.0	47.0
10231	10.2	38.0	16.0	38.0
10232	10.6	53.0	17.0	46.0
10233	11.7	70.0	20.0	48.0

N = 28

STUD. NO.	V A R I A B L E ARITH	ATPP	FINAL	ATTIT
10301	9.4	42.0	4.0	34.0
10303	9.4	39.0	6.0	38.0
10304	6.7	41.0	8.0	37.0
10305	5.8	11.0	5.0	35.0
10306	8.6	25.0	11.0	56.0
10307	10.4	36.0	4.0	47.0
10308	11.0	32.0	11.0	43.0
10309	10.4	47.0	8.0	36.0
10310	9.2	37.0	6.0	44.0
10311	8.5	11.0	11.0	49.0
10312	8.3	41.0	7.0	41.0
10313	8.1	31.0	9.0	36.0
10314	10.5	35.0	6.0	46.0
10316	11.1	65.0	10.0	40.0
10318	9.2	40.0	2.0	45.0
10319	9.1	32.0	13.0	43.0
10320	9.2	31.0	6.0	34.0
10321	9.8	17.0	7.0	26.0
10322	9.2	47.0	8.0	38.0
10323	10.8	40.0	12.0	49.0
10324	9.2	40.0	8.0	42.0
10325	9.2	44.0	9.0	38.0
10327	8.9	45.0	6.0	47.0
10328	8.1	45.0	10.0	51.0
10329	8.9	35.0	6.0	36.0
10330	10.3	59.0	11.0	30.0
10331	8.9	30.0	8.0	34.0
10332	10.0	41.0	8.0	56.0

N = 28

STUD. NO.	V A R I A B L E ARITH	ATPP	FINAL	ATTIT
10401	10.4	34.0	15.0	48.0
10402	10.3	51.0	11.0	48.0
10403	8.0	43.0	11.0	41.0
10404	8.3	48.0	10.0	37.0
10405	9.3	35.0	14.0	47.0
10406	9.6	41.0	14.0	46.0
10407	10.8	47.0	17.0	49.0
10411	9.1	38.0	12.0	46.0
10413	9.3	26.0	12.0	39.0
10414	8.3	34.0	10.0	40.0
10415	6.3	27.0	10.0	44.0
10416	6.5	28.0	10.0	35.0
10418	8.7	40.0	11.0	54.0
10419	9.1	26.0	9.0	46.0
10420	10.1	50.0	10.0	57.0
10421	9.3	41.0	11.0	49.0
10422	9.2	64.0	12.0	45.0
10424	9.4	29.0	12.0	42.0
10425	9.3	43.0	11.0	24.0
10426	9.2	35.0	9.0	39.0
10427	9.3	30.0	12.0	46.0
10428	10.3	36.0	8.0	49.0
10429	10.0	45.0	15.0	52.0
10430	9.6	49.0	13.0	48.0
10431	9.7	52.0	10.0	50.0
10433	10.1	31.0	13.0	45.0
10434	11.7	56.0	15.0	42.0
10435	9.8	36.0	10.0	39.0

N = 28

STUD. NO.	VARIABLE ARITH	ATPP	FINAL	ATTIT
10501	9.9	48.0	7.0	41.0
10502	9.0	21.0	10.0	44.0
10503	7.9	30.0	8.0	38.0
10504	8.6	33.0	9.0	46.0
10505	9.7	32.0	9.0	49.0
10506	8.1	35.0	10.0	49.0
10507	9.8	32.0	9.0	36.0
10508	7.6	26.0	8.0	43.0
10509	7.5	29.0	6.0	46.0
10510	10.0	58.0	10.0	44.0
10511	8.6	21.0	7.0	49.0
10512	8.0	19.0	4.0	48.0
10513	10.5	41.0	10.0	47.0
10514	9.1	56.0	11.0	53.0
10515	7.3	23.0	9.0	25.0
10516	8.5	37.0	10.0	36.0
10517	8.2	34.0	6.0	37.0
10518	9.4	39.0	7.0	50.0
10519	10.4	45.0	14.0	51.0
10520	11.0	48.0	8.0	36.0
10521	7.9	23.0	8.0	45.0
10522	6.9	21.0	6.0	43.0
10523	6.1	26.0	4.0	46.0
10525	5.9	32.0	13.0	49.0
10526	8.7	39.0	9.0	43.0
10527	8.7	36.0	6.0	42.0
10528	8.6	33.0	6.0	36.0
10530	9.8	22.0	7.0	52.0

N = 28

STUD. NO.	V A R I A B L E ARITH	ATPP	FINAL	ATTIT
10601	8.1	32.0	4.0	53.0
10602	8.6	53.0	12.0	50.0
10603	7.8	35.0	11.0	46.0
10604	9.4	49.0	15.0	44.0
10605	9.3	42.0	10.0	42.0
10606	9.9	40.0	15.0	44.0
10607	10.5	20.0	12.0	47.0
10608	11.0	41.0	14.0	42.0
10609	10.7	50.0	10.0	46.0
10610	8.9	47.0	12.0	51.0
10611	10.1	50.0	18.0	38.0
10612	9.4	21.0	13.0	42.0
10614	11.1	48.0	15.0	42.0
10615	9.6	52.0	12.0	54.0
10617	11.3	55.0	10.0	51.0
10618	10.4	43.0	14.0	39.0
10619	9.1	26.0	11.0	36.0
10620	8.9	48.0	16.0	48.0
10621	8.2	38.0	12.0	46.0
10622	6.0	19.0	15.0	33.0
10623	7.8	27.0	12.0	39.0
10624	10.3	38.0	14.0	42.0
10625	9.8	59.0	17.0	44.0
10626	7.9	50.0	5.0	44.0
10627	5.9	31.0	12.0	33.0
10629	8.8	36.0	9.0	49.0
10631	10.7	56.0	14.0	50.0
10632	10.4	61.0	16.0	39.0

N = 28

STUD. NO.	V A R I A B L E ARITH	ATPP	FINAL	ATTIT
10701	10.1	58.0	17.0	43.0
10702	9.9	58.0	16.0	43.0
10703	11.4	52.0	19.0	50.0
10704	13.8	67.0	16.0	54.0
10705	12.0	55.0	15.0	37.0
10706	8.8	49.0	18.0	46.0
10707	11.0	55.0	17.0	47.0
10708	12.0	53.0	15.0	41.0
10709	10.3	57.0	12.0	47.0
10710	9.3	38.0	17.0	36.0
10711	8.8	38.0	13.0	37.0
10712	11.0	72.0	18.0	45.0
10713	9.8	60.0	12.0	56.0
10715	9.9	29.0	16.0	43.0
10716	10.0	60.0	21.0	48.0
10717	10.4	39.0	11.0	51.0
10718	11.6	49.0	16.0	50.0
10719	11.1	57.0	13.0	40.0
10720	9.1	39.0	11.0	33.0
10721	8.6	36.0	16.0	39.0
10722	9.5	44.0	12.0	47.0
10723	9.8	40.0	12.0	36.0
10724	9.4	49.0	13.0	42.0
10725	8.8	43.0	10.0	35.0
10726	9.3	37.0	16.0	40.0
10727	11.4	41.0	17.0	47.0
10728	9.2	40.0	11.0	40.0
10729	9.0	54.0	14.0	42.0

N = 28

STUD. NO.	V A R I A B L E ARITH	ATPP	FINAL	ATTIT
10801	6.5	36.0	13.0	42.0
10802	7.5	44.0	17.0	41.0
10803	9.3	45.0	11.0	44.0
10804	10.8	50.0	16.0	51.0
10805	7.9	50.0	10.0	50.0
10806	11.0	50.0	12.0	47.0
10807	8.2	54.0	13.0	45.0
10808	9.0	50.0	11.0	45.0
10809	10.1	49.0	13.0	44.0
10810	11.2	52.0	14.0	49.0
10811	7.1	51.0	10.0	48.0
10812	9.4	51.0	11.0	38.0
10813	7.9	44.0	17.0	45.0
10814	9.8	51.0	12.0	38.0
10815	8.9	45.0	12.0	47.0
10816	8.7	42.0	15.0	46.0
10817	10.3	51.0	15.0	39.0
10818	10.6	66.0	20.0	52.0
10819	9.6	52.0	13.0	48.0
10820	7.3	36.0	14.0	50.0
10821	8.1	54.0	15.0	45.0
10822	10.5	60.0	17.0	45.0
10823	9.5	65.0	19.0	42.0
10824	8.6	20.0	9.0	38.0
10826	11.5	48.0	16.0	50.0
10827	9.6	45.0	16.0	45.0
10829	10.1	54.0	16.0	44.0
10830	11.1	52.0	15.0	39.0

N = 28

STUD. NO.	VARIABLE ARITH	ATPP	FINAL	ATTIT
10901	7.2	34.0	11.0	27.0
10902	9.5	43.0	7.0	51.0
10903	8.8	41.0	8.0	49.0
10904	6.9	26.0	12.0	43.0
10905	9.3	34.0	10.0	32.0
10906	9.0	39.0	5.0	51.0
10907	7.9	24.0	7.0	41.0
10908	8.8	34.0	18.0	47.0
10909	9.9	40.0	14.0	47.0
10910	11.3	39.0	14.0	42.0
10911	8.1	32.0	9.0	37.0
10912	7.2	33.0	7.0	39.0
10913	8.1	45.0	10.0	31.0
10914	8.3	39.0	10.0	42.0
10915	9.4	32.0	5.0	34.0
10916	6.4	31.0	13.0	37.0
10917	8.8	24.0	13.0	52.0
10918	8.3	35.0	10.0	44.0
10920	8.8	43.0	10.0	47.0
10922	9.2	38.0	7.0	36.0
10923	10.2	56.0	10.0	27.0
10924	11.3	37.0	10.0	37.0
10925	8.8	28.0	8.0	39.0
10926	8.8	52.0	11.0	41.0
10927	9.3	41.0	15.0	48.0
10928	11.3	37.0	10.0	38.0
10930	9.8	50.0	10.0	39.0
10931	11.1	37.0	8.0	26.0

N = 28

SELECTED BIBLIOGRAPHY

Atchison, William. "The Impact of Computer Science Education on the Curriculum". The Mathematics Teacher, 66: 1 January, 1973.

Bright, R. Louis. "There's a Computer in Your Future - The Time is Now", American Education, 3: November, 1967, pp. 12 -14.

Bucholz, Arden, Jr. "The Computer: An Evaluation", The Independent School Bulletin, 29: 27 28, December, 1969.

Computation Planning, Incorporated. "Functional Analysis and Preliminary Specifications for a Single Integrated Central Computer System for Secondary Schools and Junior Colleges", (interim report prepared for the U.S. Office of Education, Contract No. OEC-0-8-009011-2507, May, 1968), pp. 1-2.

Dwyer, Thomas A. "Soloworks: Computer-Based Laboratories for High School Mathematics". School Science and Mathematics 75: 93 - 99, January, 1975.

Fiedler, Leigh A. "A Comparison of Achievement Resulting from Learning Mathematical Concepts by Computer Programming Versus Class Assignment Approach", (unpublished Ph. D. thesis University of Oklahoma, 1969).

Foster, Thomas E. "The Effect of Computer Programming Experiences on Student Problem Solving Behaviors in Eighth Grade Mathematics", (unpublished Ph. D. thesis, University of Wisconsin, 1972).

Frye, Charles H. and Elbert C. Pack. "A Comparison of Three Computer Operating Modes for High School Problem Solving", (mimeographed report No. TM-4356/001/007, Santa Monica, California: System Development Corporation, 1969), p. 19.

Fulton, J. A. "Effects of Immediate and Delayed Information Feedback on Change Scores of Students Designated Certain and Uncertain Responses", (unpublished Ph. D. thesis, University of Illinois, 1969).

Gay, Lorraine R. "Delay of Feedback and Its Effect on the Retention of Three Types of Computer-Assisted Instruction", paper presented at the American Educational Research Association meeting, Chicago, April, 1972.

Gocker, Phillip L. "Programming Can Be Taught in the High School", Business Education Forum, 25: 58 - 59, December, 1970.

Gold, M. "Methodology For Evaluating Time-Shared Computer Usuage", (unpublished Ph. D. dissertation, Massachusetts Institute of Technology, Alfred P. Sloan School of Management, 1967).

Gumm, Robert D. "An Analysis and Development of a Computer Science Program for Use in Secondary School Mathematics", (unpublished Ed. D. thesis, Oklahoma Stated University, 1969).

Hanson, Robert D. "Machine Use in Teaching Data Processing Concepts", (unpublished Ph. D. thesis, Colorado State University, 1968).

Hatfield, Larry L. and Thomas E. Kieren. "Computer Assisted Problem-Solving in School Mathematics", Journal for Research in Mathematics Education. 3: 99 - 112, 1972.

Hoffman, Walter, et al. "Computers for School Mathematics", The Mathematics Teacher. 58: 393 - 401, 1965.

Holoien, Martin O. "Calculus and Computing: A Comparative Study of the Effectiveness of Computer Programming as an Aid in Learning Selected Concepts in First-Year Calculus", (unpublished Ph. D. thesis, University of Minnesota, 1970).

Hrasky, William C. "Remote Computing Versus the Free-Standing Computer", The Independent School Bulletin. 29: 25 - 26, December, 1969.

International Business Machines Corporation. IBM Aptitude Test for Programmer Personnel, SR 29 - 0026 - 0, 1964.

International Business Machines Corporation. IBM Scientific Subroutine, GH 20 - 0225, 1971.

Ingle, Henry T. "An Attitude Change Study on Children's Perceptions of the Computer as an Expert Source of Information", (unpublished doctoral dissertation, Stanford University, 1973).

Jacobs, Zeney, Francis G. French, William J. Moulds, Jacob G. Schuchman. Communicating With the Computer. Boston: Allyn and Bacon, Inc., 1974.

Katz, Saul M. "A Comparison of the Effects of Two Computer Augmented Methods of Instruction with Traditional Methods Upon Achievement of Algebra Two Students in a Comprehensive High School", (unpublished Ph. D. thesis, Temple University, 1971).

Kemeny, John G. "The Role of Computers and Their Applications in the Teaching of Mathematics", Needed Research in Mathematics Education. (Edited by Howard Fehr.) New York: Teachers College Press, 1970.

Kieren, Thomas E. "Computer Programming for the Mathematics Laboratory", Mathematics Teacher, 66: 1 - 9, January, 1973.

Meserve, Bruce E. "Evolution in College Mathematics", The Continuing Revolution in Mathematics. Washington, D. C.: National Council Teachers of Mathematics, 1968.

Milner, Stuart D. "The Effects of Teaching Computer Programming on Performance in Mathematics", (unpublished Ph. D. thesis, University of Pittsburgh, 1972).

Morgan, Richard T. "The Role of the Computer in a General Education Course in Mathematics", (unpublished Ed. D. thesis, Columbia University, 1968).

"Data Processing and Computer Programming Curriculum", National Business Education Quarterly. 36 (Winter, 1967), 38 - 41.

NCTM Committee on Computer Oriented Mathematics. "Computers for School Mathematics", The Mathematics Teacher. 58: 393 - 401, 1965.

Newman, Morris I., et al. "Delay of Information Feedback in an Applied Setting: Effects on Initially Learned and Unlearned Items", The Journal of Experimental Education. (Volume 42, Number 4. Summer, 1974).

Pack, Elbert C. "The Effect of Mode of Computer Operation on Learning a Computer Language and on Problem Solving Efficiency of College Bound High School Students", (unpublished Ed. D. thesis, University of California at Los Angeles, 1970).

Pellegrino, James P. "A Model for Evaluation of Disadvantaged Students Participating in a Computer Science Program", (unpublished Ph. D. thesis, University of Pittsburgh, 1971).

Reno, Martin, Charles Reno, and William Saul. "Utilization of Computer Facilities in the Mathematics and Business Curriculum in a Large Suburban High School", (report prepared for the U. S. Office of Education, Cooperative Research Project No. S-232, Euclid Board of Education and Case Institute of Technology, Cleveland, Ohio 1967).

Ronan, Franklin D. "A Study of the Effectiveness of a Computer When Used as a Teaching and Learning Tool in High School Mathematics", (unpublished Ph. D. thesis, Northern Illinois University, 1972).

Rudolph, Eleanore L. "A Survey of Data Processing and Computer Use in Instruction in Illinois Secondary Schools", (unpublished Ed. D. thesis, Northern Illinois University, 1972).

Sackman, H. and Michael Gold. "Time-Sharing Versus Batch Processing: An Experimental Inquiry into Human Problem-Solving", (report prepared for System Development Corporation, SP-3110, Santa Monica, California, 1968).

Sackman, Harold. Man-Computer Problem Solving. Princeton, N. J.: Auerbach Publishers, Inc., 1970.

Sassenrath, J. M. et al. "Immediate and Delayed Feedback on Examinations and Immediate and Delayed Retention", California Journal of Research Education. 19: 226 - 231, 1969.

Sherrill, James. "The Effects of Different Presentations of Mathematical Word Problems Upon the Achievement of Tenth Grade Students", School Science and Mathematics. 73: 4. 277 - 281, April, 1973.

Thesing, Gary L. "Some Computer Oriented Materials for Use in Elementary Calculus and an Experiment With Their Use", (unpublished Ed. D. thesis, Oklahoma State University, 1971).

Verardi, Eugene J. "Comparison of Two Methods in Utilizing the Keypunch and Computer in a Computer-Based Mathematics Course", (unpublished master's thesis University of Maryland, 1971).

Wallace, David C. "The Impact of Computer Mathematics on the Learning of High School Trigonometry and Physics", (unpublished Ph. D. thesis, University of Texas, 1968).

Weeg, Gerald. Proceedings of a Conference on Computers in the Undergraduate Curriculum. Editor's note, University of Iowa, 1970.

Wilkinson, Arthur. "An Analysis of the Effect of Instruction in Electronic Computer Programming Logic on Mathematical Reasoning Ability", (unpublished Ed. D. thesis, Lehigh University, 1972).

Winer, B. J. Statistical Principles in Experimental Design. New York: McGraw-Hill, 1962.