INFORMATION TO USERS

While the most advanced technology has been used to photograph and reproduce this manuscript, the quality of the reproduction is heavily dependent upon the quality of the material submitted. For example:

- Manuscript pages may have indistinct print. In such cases, the best available copy has been filmed.
- Manuscripts may not always be complete. In such cases, a note will indicate that it is not possible to obtain missing pages.
- Copyrighted material may have been removed from the manuscript. In such cases, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, and charts) are photographed by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each oversize page is also filmed as one exposure and is available, for an additional charge, as a standard 35mm slide or as a 17"x 23" black and white photographic print.

Most photographs reproduce acceptably on positive microfilm or microfiche but lack the clarity on xerographic copies made from the microfilm. For an additional charge, 35mm slides of 6"x 9" black and white photographic prints are available for any photographs or illustrations that cannot be reproduced satisfactorily by xerography.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Order Number 8717204

A comparative analysis of the guided-discovery method versus the traditional lecture-laboratory method in teaching introductory computer science

Howerton, Charles Paul, Ph.D.

University of Denver, 1987

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

PLEASE NOTE:

In all cases this material has been filmed in the best possible way from the available copy. Problems encountered with this document have been identified here with a check mark __√__.

1. Glossy photographs or pages _____
2. Colored illustrations, paper or print ______
3. Photographs with dark background _____
4. Illustrations are poor copy ______
5. Pages with black marks, not original copy ______
6. Print shows through as there is text on both sides of page ______
7. Indistinct, broken or small print on several pages __✓__
8. Print exceeds margin requirements ______
9. Tightly bound copy with print lost in spine ______
10. Computer printout pages with indistinct print ______
11. Page(s) __________ lacking when material received, and not available from school or author.
12. Page(s) __________ seem to be missing in numbering only as text follows.
13. Two pages numbered ______. Text follows.
14. Curling and wrinkled pages _____
15. Dissertation contains pages with print at a slant, filmed as received ________
16. Other____

University
Microfilms
International

A COMPARATIVE ANALYSIS OF THE GUIDED-DISCOVERY METHOD VERSUS THE TRADITIONAL LECTURE-LABORATORY METHOD IN TEACHING INTRODUCTORY COMPUTER SCIENCE

A Dissertation

Presented to

The Faculty of Social Sciences

University of Denver

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

---

by

Charles Paul Howerton

June 1987

GRADUATE STUDIES

AT

THE UNIVERSITY OF DENVER

Upon the recommendation of the Director of the School of Education this dissertation is hereby accepted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

William M Slaichert

Professor in charge of dissertation

Dean of Graduate Studies

June 1987

Date

Table of Contents

Chapter Page

APPENDICES

List of Tables

Table Page

Table Page

Table Page

Table Page

Acknowledgments

I would like to express my sincere appreciation to Dr. Jean Bell, Mr. Patrick Madison, and Dr. Donald Marsh of the faculty of Mathematics and Computer Science of Colorado School of Mines and Dr. Ardel J. Boes, their chairperson for their encouragement, cooperation and participation in this study. I would also like to thank the students of the EPICS program and the three sections of MA-115 for their participation. Finally, I would like to thank two members of the faculty at The University of Denver, Dr. William M. Slaichert of the School of Education and Dr. Ruth Hoffman of the Department of Mathematics and Computer Science for their encouragement, patience and support throughout this effort.

CHAPTER 1

INTRODUCTION

BACKGROUND OF THE STUDY

Few advances in technology have aroused the interest and attention of the general populace to the same extent as the modern personal computer. The advent of the microcomputer and its related technology has made it possible for every individual to have access to computational capability beyond that available to all but the Federal Government and a few major corporations a mere 25 years ago.

As a consequence of this unprecedented technological advance the demand for education and training in the use of computers is growing at an even greater pace as people strive to make effective use of the new technology. In response to this demand for training, Ryder (1984) has observed that

> Computer science departments currently face an overwhelming demand from the university community for computer literacy service courses.

Martin & Martin (1986) in a paper which discusses typical computer literacy students state,

> In many, if not most colleges and universities today, a computer literacy course for "noncomputer" majors is offered. Frequently the course is the only one taught by the department of computer studies that these students are required to take.

The issue of introductory computing courses at technical and engineering institutions is even more important since their students and graduates frequently make extensive use of computers as they practice their professions. Ayen and Grier (1983) addressed this issue in a paper describing the introductory computer science course offered at the United States Air Force Academy,

> The Air Force Academy requires each individual who graduates to complete an extensive curriculum oriented towards engineering and including courses in other standard academic disciplines. An introductory course in computer science is one of the required courses.

Inspection of the catalog of almost every engineering school will reveal that an introductory computer science course is required of all students.

From the foregoing, one could easily conclude that the computer science departments of most colleges and universities would be overjoyed with the opportunity to generate so many student-credit-hours, but this is not always the case. As with most rapidly growing high technology disciplines, finding adequate faculty to teach these service courses is a major problem. The demand for trained computer scientists by industry and government is overwhelming.

In the latest in a series of articles entitled "Production and Employment of Ph.D.s in Computer Science" (Taulbee, 1986; Taulbee and Conte 1979, 1977, 1976), Orrin E. Taulbee, Chairperson of the Association for Computing Machinery Computer Science Employment Register, reports that the production of Ph.D.s in the various categories of Computer Science in the decade ending June 1985 was 2,459. This is a mean annual production of 246. The lowest production of 208 was in 1977, and the highest production of 295 was in 1985 (see Table 1). During that same period employment of new Ph.D.s by colleges and universities was 946 or less than 40 percent of the graduates. This is a mean annual production of doctoral level computer science professors of 95. The lowest number entering the teaching profession during the ten year period was 81 in 1984 (84 in 1985), and the highest number was 106 in 1977. The percentage of new Ph.D.s being employed by colleges and universities is steadily declining (see table 2). In 1985, there were 2,588 computer science Ph.D. students or almost 9 times as many as were

TABLE 1

Production of New Computer Science Ph.D.'s

1976	1977	1978	1979	1980	1981	1982	1983	1984	1985
246	208	223	248	230	235	244	256	274	295

TABLE 2

Employment of New Computer Science Ph.D.'s
by Organization Type

	1976	1977	1978	1979	1980	1981	1982	1983	1984	1985
Industry	94	75	90	93	94	96	101	129	138	157
Government	9	12	11	12	10	9	11	8	9	10
College/Univ	100	106	100	103	101	97	91	83	81	84
Other	16	2	16	19	17	16	13	12	11	10

graduated, and over 30 times as many as entered the teaching profession.

The reason for this migration away from teaching into industry and government is salary. The average starting salary for new assistant professors in computer science at 262 public institutions as reported in the May 14, 1986 issue of The Chronicle of Higher Education is $29,692 per year (Evangelauf, 1986). This salary ranks third behind business with a range of staring salaries from $30,458 to $32,759 depending upon the specialty, and engineering of all types with an average starting salary of $31,302. In comparison, the average starting salary for new assistant professors in all major fields is $25,122.

In contrast, a brand new Ph.D. Computer Scientist can enter the federal government as a GS-14 at a salary in excess of $40,000 per year. In private industry, salaries of $50,000 to $60,000 per year are common knowledge which can be verified by reading help wanted advertisements in computer and electronics industry publications as well as the Wall Street Journal, The New York Times, and various other periodicals of national stature. In addition, campus recruiting by government and private industry is very aggressive with promises of significant support for research projects and for participation in professional activities and organizations. Clearly, few colleges or universities can afford to compete with these salaries and benefits, and in this day and age of the Yuppie ethic, the prestige that was once associated with being a college professor is poor compensation for the salary difference.

To compensate for the lack of educationally qualified computer scientists most colleges and universities utilize faculty from other departments to teach some of the computer science courses. In Taulbee's most recent report (Taulbee 1986), he shows that for the 86 institutions reporting faculty constituency for the past ten years, on the average only 48.3-percent of the faculty teaching computer science had doctoral degrees in computer science. It is only during the three years beginning in 1982, that the percentage has risen above 50-percent. In 1985, 52.8-percent of the faculty had Ph.D.s in Computer Science. The accepted norm for academic credentials required for teaching positions at many smaller schools has become a Master's Degree in Computer Science, or a Ph.D. in a related field such as Computer Science

Education, Numerical Analysis and other computation heavy disciplines. The average starting salaries reported in The Chronicle of Higher Education (Evangelauf, 1986) include individuals who do not have doctoral level degrees.

STATEMENT OF THE PROBLEM

One of the most difficult problems facing college and university administrators in this age of rising costs and declining enrollments is how to educate students more effectively. Most specifically, they must respond to an increasing demand for introductory courses in highly technical disciplines such as computer science and engineering, which must be taught by very highly paid faculty members.

If the administrators do not adequately respond to the demand for introductory high-technology courses, the students are likely to seek a school where such courses are offered. This will compound the problem by reducing the enrollment even further and consequently the tuition income will be reduced as well as legislative funding in the case of most publicly supported institutions.

Large schools with an active graduate program in computer science can usually fulfill the requirement by assigning graduate teaching assistants to teach the introductory courses to earn their stipend. Smaller schools and those without an adequate graduate program cannot economically assign a regular computer science faculty member to teach these courses unless there is some way to increase the productivity. There are several ways to increase productivity of faculty members. Two of the most commonly used methods are increasing class size or reducing the number of earned contact hours.

Unfortunately, learning to use a computer is much like learning to play the piano. It cannot be done in a lecture hall, it must be done at the keyboard. Thus simply increasing the number of students in a class does not solve the problem unless computer laboratories are available with a minimum of half as many keyboards as there are students in the class. The alternative technique for increasing productivity by reducing the number of earned contact hours can only be applied in a limited way under normal circumstances. Typically, laboratory classes are credited at the rate of one earned contact hour for every three laboratory hours. However, the laboratory instructor usually acts as a supervisor and information source, and not as a lecturer on what to do and how to do it.

Clearly, what is needed is a new innovative approach to teaching these courses which retains the effectiveness of the more traditional methods while doing the job more productively. This in turn creates additional problems relating to the quality of the training associated with any specific pedagogic technique. The quality issue resolves itself into two distinct sub-issues; actual measurable quality and perceived quality. These two sub-issues have both separate and interactive effects on the student and can affect the student's continuing educational experience and employment potential. For example, if it has been established by whatever means that the students of an institution which makes use of innovative pedagogic techniques do not learn as much or as well as those who attend similar institutions which practice more traditional techniques, the students may experience difficulty in getting jobs regardless of their academic ability. Alternatively, should other schools refuse to allow transfer credit for certain courses because they perceive that the manner in which they have been taught is inadequate or incompatible with their methods, the students may be limited in their ability to pursue educational alternatives.

In the final analysis, there is only one effective manner in which to measure the quality of an innovative pedagogic technique in a particular educational setting. That is to implement the technique and measure the results. The customary manner of testing an educational technique is to apply it in a traditional experimental-group, control-group environment and compare the results by administering some form of achievement test. This method is capable of providing some qualitative measure in quantitative terms for specific groups of students. If, in the testing process, one or more questions are included which solicit the participants opinion of the relative merit of the treatment methods, it is possible to develop some concept of the student's perception of the quality of the training they received. Data gathered in this manner can provide the information needed by administrators to evaluate a potential program and to decide whether it is desirable to their institution and its clientele.

Therefore, the administrative problem of producing cost-effective education resolves itself into a pedagogic problem and a marketing problem. The pedagogic problem can be resolved by developing an acceptable method of delivery that is perceived as being equivalent in quality to the traditional methods while retaining or exceeding the effectiveness and acceptability of the traditional methods. The marketing problem requires that the institution convince potential students, other educators and future employers that the new method produces an equivalent or better educated student who should be more desirable as an employee or advanced student.

PURPOSE OF THE STUDY

The purpose of the study was to determine whether an innovative, multi-disciplinary, guided-discovery technique for teaching introductory computer science concepts would produce results equivalent to those produced in a more traditional single-subject, lecture-laboratory environment for engineering students at the Colorado School of Mines. In addition, this study was designed to determine the effect of certain demographic factors, such as prior experience with computers, on achievement in general as well as between the two teaching techniques. Finally, it was desired to determine the reactions of the students to the two teaching techniques in terms of relative educational benefit. The specific objectives of this research study were:

1. Determine if there is any significant difference in student achievement scores on a test of introductory computer science concepts as measured by posttest knowledge achievement scores when using a multi-disciplinary guided-discovery teaching technique compared to a single-subject lecture-laboratory technique.
2. Determine if there is any significant difference in student perception of educational benefit between teaching techniques as measured by a relative merit opinion score.
3. Determine if there is any significant difference in achievement scores on tests of specific subtopics of introductory computer science as measured by posttest knowledge scores when using a multi-disciplinary, guided-discovery teaching technique compared to a traditional single-subject lecture-laboratory technique. The subtopics are:

- (1) General FORTRAN Knowledge,
- (2) Functions and Procedures,
- (3) Input-Output Statements,
- (4) Program Control Statements,
- (5) Reading and Understanding Programs,
- (6) Miscellaneous Computing Concepts,
- (7) Expressions, and
- (8) Arrays.

4. Determine if there is any significant difference in student achievement scores on a test of introductory computer science concepts and on the specific subtests as measured by posttest knowledge scores when comparing demographic characteristics. The specific demographic characteristics to be analyzed are:
    - (1) Sex of student,
    - (2) Computing Courses in High School
    - (3) Computer Use Prior to College
    - (4) Ownership of a Personal Computer

At the beginning of the study the departments of Engineering, English, and Mathematics and Computer Science at Colorado School of Mines were just beginning the second phase of a pilot program to test the feasibility of an integrated course which would incorporate introductory concepts from the three disciplines. The first phase had been completed the previous year.

The first phase of the pilot program had been conducted by one faculty member from each of the participating departments. Its purpose had been to develop the expertise of the participating faculty with the teaching technique. Thirty hand-selected student volunteers participated in the first phase.

The second phase of the pilot program was intended to be a large scale test of the technique on a significant percentage of the entering freshman class. The purpose was to further refine the teaching techniques, and test them on a more representative student sample. The 120 participating students were selected randomly from approximately 240 volunteers.

While phase two of the pilot test was being conducted with approximately one-third of the entering freshmen, it was necessary to continue offering the traditional courses in all three disciplines. The researcher was one of the three professors who were chosen to teach the same computer science material in the traditional mode.

HYPOTHESES TO BE TESTED

Hypothesis 1

There will be no significant difference between the mean achievement scores for the Total Test when the subjects are differentiated by Teaching Method. The Teaching Methods used were the Guided-Discovery method for the Experimental Group, and the traditional lecture-laboratory method for the Control Group.

Hypothesis 1.1

There will be no significant difference between the mean achievement scores for the General FORTRAN Knowledge Subtest when the subjects are differentiated by Teaching Method.

Hypothesis 1.2

There will be no significant difference between the mean achievement scores for the Functions and Procedures Subtest when the subjects are differentiated by Teaching Method.

Hypothesis 1.3

There will be no significant difference between the mean achievement scores for the Input-Output Techniques Sub-Test when the subjects are differentiated by Teaching Method.

Hypothesis 1.4

There will be no significant difference between the mean achievement scores for the Program Control Statements Subtest when the subjects are differentiated by Teaching Method.

Hypothesis 1.5

There will be no significant difference between the mean achievement scores for the Reading and Understanding Programs Subtest when the subjects are differentiated by Teaching Method.

Hypothesis 1.6

There will be no significant difference between the mean achievement scores for the Miscellaneous Computing Concepts Subtest when the subjects are differentiated by Teaching Method.

Hypothesis 1.7

There will be no significant difference between the mean achievement scores for the Use of Expressions Subtest when the subjects are differentiated by Teaching Method.

Hypothesis 1.8

There will be no significant difference between the mean achievement scores for the Use of Arrays Subtest when the subjects are differentiated by Teaching Method.

Hypothesis 2

There will be no significant difference in the REACTION to the Teaching Method Opinion Ratings of the subjects when differentiated by Teaching Method they received.

Hypothesis 3

There will be no significant differences between the mean achievement scores for the Total Test when the subjects are grouped by the TEACHER who taught them.

Hypotheses 3.1 Through 3.8

There will be no significant differences between the mean achievement scores for each of the eight subtests when the subjects are grouped by the TEACHER who taught them.

Hypothesis 4

There will be no significant difference between the mean achievement scores on the Total Test when the subjects are grouped by their SEX.

Hypotheses 4.1 Through 4.8

There will be no significant differences between the mean achievement scores on each of the eight subtests when the subjects are grouped by their SEX.

Hypothesis 5

There will be no significant difference between the mean achievement scores on the Total Test when the subjects are grouped by whether they took any COMPUTING COURSES in HIGH SCHOOL.

Hypotheses 5.1 Through 5.8

There will be no significant differences between the mean achievement scores on each of the eight subtests when the subjects are grouped by whether they took any COMPUTING COURSES in HIGH SCHOOL.

Hypothesis 6

There will be no significant difference between the mean achievement scores on the Total Test when the subjects are grouped by whether they had MADE REGULAR USE OF A COMPUTER PRIOR TO COLLEGE.

Hypotheses 6.1 Through 6.8

There will be no significant differences between the mean achievement scores on each of the eight subtests when the subjects are grouped by whether they had MADE REGULAR USE OF A COMPUTER PRIOR TO COLLEGE.

Hypothesis 7

There will be no significant difference between the mean achievement scores on the Total Test when the subjects are grouped by whether they OWNED A PERSONAL COMPUTER AND PROGRAMMED IT BEFORE ENTERING COLLEGE.

Hypotheses 7.1 Through 7.8

There will be no significant differences between the mean achievement scores on each of the eight subtests when the subjects are grouped by whether they OWNED A PERSONAL COMPUTER AND PROGRAMMED IT BEFORE ENTERING COLLEGE.

DEFINITION OF TERMS

In the interest of eliminating any ambiguities which might result from dissimilar interpretation of terminology, the following operational definitions are provided for reference:

Guided-Discovery Method: An instructor guided study in which the instructor provides minimal direct instruction in the form of brief introductory lectures at the beginning of laboratory oriented sessions. The instructor acts as a resource to the students and also provides guidelines for studying and deadlines for assignments. The student has direct responsibility for learning the material at the pace established by the instructor.

Lecture-Laboratory Method: A traditional instructional method wherein the instructor lectures or uses other classroom oriented techniques for one or more "academic" hours per week. This instruction is typically associated with a separate laboratory session taught by a laboratory instructor who supervises a two or three hour "hands-on" session in which the students apply what they have been taught.

CHAPTER 2

REVIEW OF LITERATURE

The literature search for this review included articles published in educational, and computer science journals over the past five years. In addition, computerized searches were performed on various data bases contained in the Lockheed Dialog System including Dissertation Abstracts, ERIC, and others containing computer science and education related material.

Four topics are addressed specifically in the review of the literature. First those items relating to the stated problem directly are discussed in the Background of the Study. The next three sections, Alternative Teaching Techniques, Alternative Solutions to the Faculty Shortage Problem and Alternative Solutions to the Number of Students, are a review of the literature related to the problem.

Background of the Study

Dubin and Taveggia (1968) have performed an analysis of over 90 studies on comparative teaching methods in which the principle focus was to improve the educational benefit to the student. They stated their results in quite unambiguous terms on page 8.

> The evidence is all in upon which we may base our conclusions about the relative utility of given methods of college teaching, when this utility is measured through final examinations: THERE ARE NO DIFFERENCES THAT AMOUNT TO ANYTHING.

On page 35 they amplify their findings with specific references to the scope of their study:

> ... we have reported the results of a reanalysis of the data from 91 comparative studies of college teaching technologies conducted between 1924 and 1965. These data demonstrate clearly and unequivocally that there is no difference among truly distinctive methods of college instruction when evaluated by student performance on final examinations.

Finally in their conclusions, on page 45, they leave no doubt about their feelings:

> We have found no shred of evidence to indicate any basis for preferring one teaching method over another as measured by the performance of students on course examinations.

Even though the Dubin and Taveggia study is over twenty years old, very little has changed. College and university faculty and administrators continue to search for an instructional method which will deliver more education for the same effort.

It is not very likely that Dubin and Taveggia included any studies which were computer science related since the earliest comparative study in computer science was done in 1965 by Dale J. Hall (Hall 1965). Hall asserts in his report that he found no earlier studies on "the teaching of FORTRAN computer programming by programmed instruction." However, Dubin and Taveggia would not have been surprised at the results as reported by Hall.

> One analysis of variance and four analyses of covariance were computed for the FORTRAN Facts Test Data. Results of these analyses supported Hypothesis I: there will be no significant differences in achievement among the four experimental groups as measured by a test covering the basic FORTRAN facts.
>
> The same analysis procedure was followed for the Problem Evaluation Test Data. Results of the analysis supported Hypothesis II: there will be no significant difference among the four experimental groups as measured by a problem evaluation test.

A review of Hall's bibliography reveals citations for numerous studies in other disciplines but none in computer science. Thus, one can conclude that the first comparative teaching methods study in computer science produced essentially the same results as previous studies in other disciplines. Numerous other computer science teaching methods researchers since Hall have also found that the method makes very little difference in educational benefit to the student (Calamari 1983, Clark 1975, Drew & Caplin 1984, Grossman 1983, Martin 1985, McLaughlin 1981, Miller 1981, Narthasilpa 1984, Payne 1983, Pommersheim 1983, Reisman 1973, Rusnock 1983, Thronson 1984, Wiggins

1984, Witherell 1979). One researcher (McEntyre 1977) reported that in a restructured course the students performed significantly better on the mid-term and final examinations.

Since teaching method does not seem to make any significant difference to student achievement as measured by final examination scores then perhaps the teaching method used is not significant to the student. If this is true, why should educators continue to perform comparative analyses of various teaching techniques? There are at least two answers to this question, and both are provided by Dubin and Taveggia in their conclusions. The first is discussed on page 46 and suggests a whole new direction for comparative teaching methods research.

> Future research on comparative teaching methods must focus on the question: "What is there that is the same about any two different teaching methods?"

The second reason for continuing to evaluate different teaching methods is to perform cost-benefit analyses. Once again, Dubin and Taveggia (1968:49) point the way:

> Increasing attention will be demanded of college and university administrators to the cost-benefit analysis of various teaching methods Up to this point, the "benefit" portion of cost-benefit analysis has largely depended upon private opinion and prejudice. We think that we have demonstrated in this monograph that the usual

> prejudices regarding preferred college teaching methods are no longer acceptable as bases for alleging the benefits of particular teaching technologies.
>
> Indeed, since there are no differences among a wide range of teaching technologies we may assume that their respective benefits are equal. This, then, turns the attention in cost-benefit analysis to the cost side of the issue.
>
> In making the costing decisions the obvious strategy would seem to be to pay out as little as possible for instructional costs.

The discussion then goes on to discuss some of the more common methods of reducing costs or increasing faculty productivity.

Alternative Teaching Techniques

A taxonomy of teaching techniques would probably be an endless list consisting of uniquely identified variations of combinations and permutations of a small well defined set of methods. Several researchers have attempted to identify and enumerate teaching strategies and techniques.

Dubin and Taveggia (1965) identified 91 studies comparing teaching methods and techniques which had been performed during the period from 1925 to 1965. They identified two generic techniques, face-to-face instruction and independent study. By their definition, face-to-face instruction includes all of the variations of lecture, group-discussion and the tutorial. Dubin and Taveggia identify two categories of independent study methods which are, in reality, simply the two extremes on a continuum from closely-supervised teacher-directed and guided independent study to totally unsupervised independent study.

James R. Davis in a monograph entitled Teaching Strategies for the College Classroom (Davis 1976) has reduced the list of "standard" strategies to four generic categories. The categories identified by Davis are (1) instructional systems, (2) lectures, (3) inquiry, and (4) group processes. Variations on these four basic themes are discussed in depth within the monograph. Davis' definition of teaching strategy is applicable regardless of the motive for adopting the strategy.

> Applied to college teaching, the term strategy refers to a plan, method, or series of activities designed to achieve a particular educational goal.

Davis, too, concedes that most studies which compare teaching strategies find, what he calls the researcher's anathema, "no significant difference." However, he points out that there is a time and a place for each of the teaching strategies, and that consistant application of a strategy is most important.

Davis reports that one researcher has questioned the effectiveness of the traditional methods.

> Ohmer Milton makes the case that the effectiveness of traditional methods of instruction at the college level has never been proven. Traditional methods are vigorously supported, yet there seems to be little concrete evidence to warrant such enthusiasm.
>
> The studies Milton has drawn together demonstrate that the current research on college teaching fails to lend much support for continuing to use traditional teaching methods. Unfortunately, there is not much evidence that alternative methods are much better.

The message here seems to be that teaching method may not have much to do with learning. Which, in turn, challenges the traditional assumptions about the role of the teacher in the learning process.

Since the advent of the modern electronic computer, is has been the goal of some educators to convert it into the ultimate teaching machine. A recent edition of "Electronic Education," a periodical catalog devoted to promoting the microcomputer as an educational tool contained advertisements for literally hundreds of computer aided instruction programs. Most of these programs were designed to teach some subject to a specified audience utilizing a particular microcomputer. Wiggins (1984) used microcomputer-assisted instruction as an alternative method for teaching the programming language BASIC to Agricultural Engineers at Iowa State University. He reports:

> An analysis of covariance failed to reveal a significant difference in either posttest attitude scores or posttest knowledge scores when the students were grouped by teaching method (computer-assisted instruction versus traditional lecture).

Wiggins then observes that "computer-assisted instruction is neither superior nor inferior as a teaching method to traditional lecture."

In a study which compared independent learning with teacher-guided learning involving formal human instruction and use of a canned instructional program, Narthasilpa found that there were no significant differences in the mean scores of the two groups on four different criteria (Narthasilpa 1984).

Self-instructional texts are a common method for teaching some computing topics. Usually, use of the self-instruction text is accompanied by interactive sessions on the computer. However, at the time of the earliest study of an alternative method for teaching computer programming (Hall 1965), interactive computing facilities were not readily available. Hall designed a course to teach the principles of FORTRAN programming to high school students participating in a special summer program at Indiana University. The course was taught through the use of "tape recorded lectures with accompanying visual presentations and student workbooks." The experiment was performed with three groups which tried three different combinations of the materials and a control group which used a programmed instruction text from International Business Machines. Hall found no significant differences between either the attitude or achievement scores of the four groups.

In 1981, McLaughlin compared discovery learning with programmed instruction in an attempt "to improve the usefulness of discovery learning with both methodological and substantative contributions." Three levels of discovery "differing primarily in the amount of personal responsibility and initiative required of the learners" were designed as experimental treatments. The control treatment was the use of a Computer-Assisted Instruction program which covered the same material. McLaughlin performed several tests for significant differences in the treatments both between the experimental groups and between the experimental groups and the control group. He found a non-significant tendency for the two discovery groups which required the most initiative to score higher on some areas and to score significantly higher in others.

However, in comparison of combined discovery performance versus programmed instruction performance there was no significant difference due mainly to the poor performance of the low-initiative discovery group.

Self-paced instruction learning has been used in almost every discipline, and computer science is no exception. Several studies have been done on self-paced learning (Narthasilpa 1984, Witherell 1979) Witherell performed a similar experiment four years later (Witherell 1979) at Southeastern Massachusetts University on Freshmen enrolled in the College of Business and Industry. Witherell created three groups, two control groups and one experimental group. The two control groups were taught by the traditional lecture method, with one group meeting three times a week for 50 minutes and the other group meeting twice a week for 75 minutes. The experimental group class met twice a week for 75 minutes, but attendance was not compulsory, and the teacher acted in a tutorial and resource capacity. Experimental group subjects were allowed to advance at their own pace with no controls. They were not required to master each topic before advancing to the next. Little or no control was exercised over the control group students. Witherell found that the self-paced students achieved as well as the traditional students.

Other studies which are not so easily categorized have repeatedly found that there are no significant differences in achievement scores between the experimental groups and the control groups. The variety of treatments which have been attempted defy categorization. Drew and Caplin (1984) redesigned a class to take advantage of some cheaper resources and found that there were no significant differences in performance. Martin (1984) combined two sections of the same class, which happened to be scheduled consecutively into one larger combination lecture-laboratory section. The purpose was to try to cover more material during the semester. In this case the students in the experimental group performed worse than the previous year's students on some of the tests. Martin concluded that there was a limit to the amount of material which can be absorbed in real time. Harry E. Payne (1983) tested three different techniques for teaching verbal problem solving. The three techniques involved the application of three different problem solving techniques, specifically flow-charting, heuristics, and structured questioning. He found no significant differences between groups on posttest and retention tests. Payne concluded that any of the three methods can be used with equal effectiveness. Grossman (1983) tried two different methods of group programming teams in comparison with the traditional individual programming method. She found that the experimental groups produced more homework of higher quality on time, but there were no significant differences in measures of programming proficiency or efficiency. Mary Calamari compared the effect of teaching high school algebra and geometry students LOGO prior to teaching them BASIC versus a BASIC only approach. Using an analysis of covariance with a pretest score as the covariate she found that the students taught with the BASIC only technique scored significantly higher on the criterion score. However, since this experiment was done in a severely reduced time frame it is likely that learning two programming languages rather than one accounts for the difference. Clark (1975) integrated pedagogy into a course designed to teach computer programming to elementary school teachers. The control group were taught the course in the normal manner. The experimental group was shown examples of lesson plans and instruction sequences for sixth grade students.

They were also shown video tapes of sixth grade classes being taught the material. There were no significant differences in posttest scores between the control group and the experimental group, but there was a significant difference on retention tests administered later in favor of the experimental group. Clark concluded that the method was worth using in order to improve retention. Sorel Reisman (1973) attempted to answer the question: "Is there a difference in achievement in programming between groups which learn a high-level language first followed by a low-level language versus the alternate sequence." No significant differences were found.

When it comes to comparing innovative or simply different teaching strategies and techniques with more traditional methods and techniques, it is not uncommon to find a positive non-significant difference in favor of the experimental method. In an information analysis report done for the Elgin Community College in Elgin Illinois, Ed Haring (Haring 1985) makes the following observation in the conclusion to the report:

> Students' learnings may be enhanced and their level of achievement increased through changes in instruction, if these changes are not what's fashionable or "in" at the time. The changes must be within the ability of the teacher and coupled

> with a real willingness and desire to enhance student achievement. The changes must take into consideration how students learn, i.e., their learning style.

Clearly, what Haring has identified here is a corollary of the Hawthorne Effect. The students' learning may be enhanced because they are interested or intrigued by the changes in instruction, or by the extra attention which is associated with being part of an experiment. This, in turn, heightens their awareness of what is happening in the classroom. What Haring does not do is describe at what point the new technique becomes the "in" thing and loses its power to stimulate.

Alternative Solutions to the Faculty Shortage Problem

One of the most important issues in computer science education today is how to solve the faculty shortage problem. The number of new computer science Ph.D.s entering the teaching profession has remained fairly constant for the past 15 years, while the number of computer science students has been growing steadily. New graduates with baccalaureate degrees are demanding and receiving salaries in excess of those paid to faculty members in most disciplines. They are accepting high paid jobs and not continuing on into graduate school. The bulk of the Master's degree students are non-traditional students on an upwardly mobile career path. They are seeking degrees for advancement in their job. Where, then, will new computer science faculty come from? How will they be paid?

It is a verifiable fact that computer science faculty can and do demand higher salaries. This has an adverse effect on other faculty members who are not fortunate enough to be in a high demand field. The solution to this problem is to retrain existing faculty members in related disciplines to teach computer science courses and then pay them appropriately. Numerous recent proposals advocate this solution (Appleby 1986, Ballew 1985, Heeler 1983, Karcan 1984, Mitchell 1983 and 1986, Mitchell and Hartman 1985, Scanlan 1985, Schmalz 1985).

At the annual meetings of the Special Interest Group for Computer Science Education of the Association for Computing Machinery for the past several years there have been special sessions devoted to the discussion of retraining of faculty to teach computer science. Computer science courses have been taught by members of other departments ever since the courses were first offered. However, this did not usually result in any salary differential for the faculty member from the other department because the faculty member did not have the appropriate credentials to warrant it. This new approach is designed to provide those credentials.

According to Mitchell and Hartman (1985), four major institutions are currently offering retraining programs which are designed to fill the need. Memphis State College, The University of South Carolina at Columbia, Clarkson University and the University of Evansville all offer programs leading to a Master's Degree in Computer Science for college and university educators wishing to cross-train. Other institutions are offering similar programs for secondary school teachers who wish to upgrade their skills.

Alternative Solutions to the Number of Students Problem

When it becomes impossible to acquire the resources to perform a specific function, such as teaching computer science courses, an alternative solution is to reduce the demand. Computer science is not an easy discipline in which to acquire a degree. The fallout rate is very high during the first two years. Several recent studies have attempted to identify those factors which are good predictors of success in computer science (Gathers 1986, Plog 1980, Ramberg 1986, Ricardo 1983, Whipkey 1984).

Plog found no significant relationship between a score on an aptitude test and academic success in an introductory computer programming course. Ramberg discovered that the single most important key to success in computer science courses is previous exposure to computers. A significant relationship between SAT Math scores and success in computer programming courses has been found by several researchers. On the other hand Gathers found that ACT English scores were the best single predictor, and that verbal scores were as important as mathematics scores to computer science success.

Catherine Ricardo examined numerous variables which appeared to be related to success in a first course in computer programming. In addition to using the SAT-Math and SAT-Verbal scores as predictors she tested deductive reasoning, persistence and inductive reasoning for relevance as predictors. The criteria were final examination grade and final course grade. She found that all five factors are significant predictors of success in a first course in computer programming.

In other studies, Rusnock (1983) found that a student's cognitive profile type as defined by Lietteri (Type 1, 2,or 3), had a significant effect on students' success in programming based on programming tests at both the comprehension and the creative level.

Thronson (1984) tested to discover if there was any significant difference in achievement in a beginning computer programming class based on learning style. He found that achievement is not a function of learning style. He also found that demographic factors were better predictors than learning style.

CHAPTER 3

METHODS AND PROCEDURES

INTRODUCTION

This study was undertaken to determine whether an innovative, multi-disciplinary, Guided-Discovery instructional environment is equivalent to a more traditional, single-subject, Lecture-Laboratory environment for teaching introductory computer programmming concepts to undergraduate engineering students at Colorado School of Mines. Equivalence was to be measured by a post-test achievement score and a student reaction opinion.

This chapter contains a description of the methods and procedures used in this study to evaluate the two teaching methods, the students reactions to the teaching methods, and several demographic factors. First, the population and sample of subjects used in the project are discussed. This is followed by a discussion of the research design, and the method of selection of the subjects. Next, the teaching methods are compared and contrasted. Then the test instrument is described and validated. Finally, the method of data collection is described in detail.

POPULATION AND SAMPLE

The general population for this study was the undergraduate student body of the Colorado School of Mines. Students at all educational levels within the school participated in the experiment as either experimental subjects or as test instrument validity controls. The target sample was those students who entered the school as freshmen at the beginning of the Fall Semester of the 1984-1985 academic year, and enrolled in either EP-101 Engineering Practices Introductory Course Sequence (EPICS) or MA-115 Introduction to Computer Programming.

All entering freshmen, nearly 400 students, were solicited by mail to participate in the EPICS experimental program. Over 240 of those solicited volunteered to participate. Of those who volunteered, 110 were selected from the pool of volunteers by random drawing to be enrolled in the EPICS program. An additional 10 participants were chosen at random from a sub-group consisting of foreign students. The remaining 270 freshmen enrolled in one of three sections of MA-115 with selection of section being governed solely by the vagaries of the normal registration process.

RESEARCH DESIGN

A post-test only design was used for this study. The participants were partitioned semi-randomly into the experimental group which would be taught using the Guided-Discovery method, and the control group which would be taught by the traditional Lecture-Laboratory method.

A large sample of the Fall 1985 entering Freshmen was also tested to provide a lower bound score for the test instrument. In addition, a small group of upper division students were also tested for an upper bound.

In this study, it has not been possible to have the level of control usually recommended by educational researchers such as Borg and Gall (1983:670). For example, selection of the subjects for the two groups was at best semi-random. The subjects for the experimental group were chosen at random from a large group of volunteers. The subjects for the control group were simply those who were not selected from the pool of volunteers plus the remainder of the population. Borg and Gall, on page 671, discuss some of the disadvantages of not administering a pretest. They point out that the lack of a pretest eliminates any possibility of controlling for initial differences between the groups. In this study, while there was no pretest based upon the test instrument or the specific subject matter, most of the subjects in the sample had taken either the Scholastic Aptitude Test (SAT) or the American Collegiate Testing Service Test (ACT). In a recent paper entitled "Predicting Student Performance in a Beginning Computer Science Class," Laurie Honour Werth from the Computer Science Department of The University of Texas at Austin reported:

> A number of studies have attempted to predict success in computer science courses and several pretests have been tried. Factors examined generally include age, gender, major, classification, college GPA and number of mathematics courses. Recently, several studies have concentrated on data available before students enroll which can be used to effectively place students in the correct computer science courses or determine which students will persist in the major. High school GPA, and number of classes or grades in high school mathematics, science, computer science and English classes, as well as SAT or ACT scores have been the factors examined. SAT and ACT scores, numbers of mathematics and science classes, and gender have been shown to be useful predictors. [Werth 1986:138]

Although the SAT and ACT tests do not measure exactly the same characteristics, there are established methods for deriving an equivalent value for analysis purposes. Thus the derived equivalent scores from these tests can be used as a covariate in the analysis of the results of the posttest.

DESCRIPTION OF TREATMENTS OR TEACHING METHODS

The Common Ground

Both the experimental group and the control group were taught introductory computer science and computer programming concepts. Both groups used the textbook "FORTRAN/77 An Introduction to Structured Problem Solving" by V. A Dyck, J. D. Lawson and J. A. Smith [1984]. Both groups were presented chapters 1 through 6, 12 and 13. The order of presentation was not the same for both groups but approximately the same amount of time was spent on each subject for both groups.

The Control Group

The control group was divided into three sections each containing approximately 85 students. Each control group section met twice per week for a 50 minute lecture. Each control group lecture section was further divided into three laboratory sub-sections of approximately 30 students. A laboratory sub-section met once per week for two hours in a microcomputer or computer terminal laboratory. The laboratory sub-section meetings were conducted by a qualified teaching assistant with previous experience in the subject. Each control group lecture section had a different instructor/teaching assistant team. Homework assignments were developed jointly by the three instructor/teaching assistant teams on a weekly basis. Tests and examinations were also developed jointly and were administered to all three sections simultaneously.

For courses in other subjects, the members of the control group had no predetermined interaction with each other. Due to the normal scheduling of Freshman courses and the fact that most Freshmen take the same courses to a large extent, it is expected that there might be a significant amount of overlap in class membership in other unrelated subjects. Specific areas where overlap is most likely to occur are in required subjects, such as Freshman English, Calculus I, and Introduction to Engineering.

The Experimental Group

The basis of the experimental program was to develop an integrated multidisciplinary course designed to prepare the students to enter the workplace. The method of instruction for the experimental program was to be guided-discovery, wherein the instructor was to provide guidance to the students who were expected to learn or "discover" the material by reading about it and applying it to problems. To this end, three academic disciplines (Computer Science, English, and

Engineering) agreed to combine their introductory courses into one integrated course. Each department contributed an instructor and teaching assistants.

The experimental group was divided into three sections of approximately 40 students each. Each section met three times per week for two hours, once with the English instructor, once with the Engineering instructor and once with the Computing instructor. The same three instructors guided all three sections, and simply met with a different section on each of three different days. All homework assignments were developed jointly by the three instructors and each assignment was designed to require work from each of the three disciplines.

The method of instruction for the Computer Science sessions was a brief introductory commentary on the subject of the week and how it related to the current assignment. The introductory presentation was not a lecture in the sense that the lecture sessions of the control group were lectures. On the average, the length of the introductory presentation was 15 to 20 minutes. During this time the instructor might give a brief example of the material and perhaps attempt to relate it to previously covered material. At this time the students would be told which pages from the textbook to read. The students were expected to learn the material independently by reading about it and by applying it to a problem to be solved.

The introductory presentation was immediately followed by a laboratory session in which the actual work was to be performed. The laboratory sessions were conducted jointly by the instructor and the teaching assistant. The students were permitted and encouraged to ask questions of both the instructor and the teaching assistant either publically or privately at any time during the two hour session. The instructor or teaching assistant would provide public or private answers depending on how the question was asked. When it was felt that the entire group might benefit from a particular response to a privately asked question it was answered publically. In these situations, both the question and the answer would be presented to the entire section.

The Instructors

The Computer Science Instructor for the EPICS group was a volunteer for the job. At the time when she volunteered she had one year of full-time college level teaching experience. The instructors for the three traditional sections were assigned based on the scheduled time for the class and their other instructional duties. Their experience levels varied from some part-time college level teaching to over 20 years of full-time college level teaching.

Treatment Summary

The two groups of subjects were presented essentially identical material from the same book over a similar period of time. The primary difference between the two groups was that the control group was taught using a traditional Lecture-Laboratory teaching method, and the experimental group was essentially self-taught using a Guided-Discovery teaching method. There were several secondary differences between the two groups. The three sections of the experimental group had only one instructor while the three sections control group had three different instructors. The order of presentation of the subject matter varied between the experimental group and the control group.

THE TEST INSTRUMENT

The test instrument was a 40 item multiple choice test containing questions directly related to the material covered in both versions of the course. The questions vary in difficulty level from very simple to extremely difficult. The content of the questions was designed to test both for general knowledge of the subject and for specific knowledge on eight sub-topics six of which are common to most programming language courses and two of which where provided to account for the questions on more general topics.

In addition to the substantive questions, several demographic questions relating to previous familiarity with computers and performance in other subjects were included. The purpose for asking the computer familiarity questions was to determine the extent to which previous experience would affect the student's success in an introductory college level course. The subjects were also asked if any other computer courses had been taken since the introductory course was taken. This question was asked so that those students who had taken other computing courses could be eliminated from the evaluation process as primary subjects. These subjects were included in the serendipitous fourth group and used as informal controls in evaluating the test instrument.

The Design of the Test Instrument

The design of the test is patterned after a similar test which is given by Heathkit Educational Systems to people who take the Heathkit FORTRAN Programming Correspondence Course. Every question was rewritten during the development of the test instrument to conform to the objectives of the course. The most significant difference between the Heathkit model test and the actual test instrument is that every question on the test instrument provides the subject with the option to answer "I Don't Know". This was included to discourage guessing by the subjects. The instructions for the test instrument specifically direct the subject to avoid guessing unless there is a high degree of confidence in the guess such as, for example, when the choice is between two similar answers.

The Content of the Test Instrument

After the researcher developed a draft version of the proposed test instrument, the four instructors were provided with copies for evaluation. The instructors met several times to provide guidance as the content of the test instrument evolved. Many of the questions were reworked several times to insure that the content was valid and that the material had been covered for all groups. Each instructor was given power to veto any question if it contained material which he or she had not covered. There were many differences in content, even among the three sections of the control group, which prevented including questions on certain topics and required that questions on other topics be reworded for clarity. The charts on the next two pages are provided to show the earliest times in the courses when the material related to the question could have been presented to the students. Finally, a last check was made to insure that every question was related to one or more of the topics in the class schedules.

In addition to insuring that all of the included questions were reasonable and proper, it was necessary to insure that all of the applicable sub-topics were adequately covered. The purpose for this was to test for breadth as well as depth of knowledge, and to provide the basis for determining whether the two teaching techniques were also equivalent for various sub-topics within the general topic. To this end,

RELATIONSHIP OF TEST INSTRUMENT QUESTIONS
BY
LECTURE, TOPIC AND CHAPTER OF THE TEXT
FOR
THE CONTROL GROUP
MA-115 Introduction to Computer Programming

LEC NO	LECTURE TOPIC	TEXT CHAP	TEST QUEST
1	Introduction	1	-
2	Problem Solving and Algorithms	2	-
3	SOS Text Editor & DEC-10 Op Sys	-	-
4	Pseudocode	2	-
5	Program Structure	2	-
6	Loops, Conditions, and Logical Ops	2	-
7	Exam One	-	-
8	The DEC FORTRAN/77 Compiler	-	13
9	Algorithms and FORTRAN/77	3	1,4,9
10	Constants/Variables/Assignment Ops	3	7,16,17,28
11	Simple I/O	3	18
12	Control Structures and Conditions	3	3,8,11,12, 19,20,21,23, 24,25,31,36
13	Data Types: Integer/Real/Character	4	34
14	Built-in Functions	4	30
15	More Advanced I/O	4	14,32
16	Midterm Exam Review	-	-
17	Midterm Exam	-	-
18	The TIPC - Hardware	-	-
19	The TIPC - Software	-	-
20	FORTRAN/77 on the TIPC	-	-
21	Subroutines	5	22,29
22	Functions	5	10,15,33
23	One-Dimensional Arrays	12	2,5,6,26,27, 35,38,39
24	Multidimensional Arrays	12	-
25	Searching and Sorting Techniques	13	37,40
26	Characteristics of Math Functions	6	-
27	Library Subroutines	-	-
28	Library Packages	-	-
29	Plotting	-	-
30	Review for Final Exam	-	-

TABLE 3

RELATIONSHIP OF TEST INSTRUMENT QUESTIONS
BY
LECTURE, TOPIC AND CHAPTER OF THE TEXT
FOR
THE EXPERIMENTAL GROUP
EP-102 Engineering Practices Introductory Course

LEC NO	LECTURE TOPIC	TEXT CHAP	TEST QUEST
0	Introduction to Semester	-	-
1	Elementary FORTRAN, Data Types	3	1,3,4,7,8,9,11,12,16,17,19,20,21,23,24,25,28,31,36
2	Arrays in FORTRAN	12	2,5,6,26,27,35,38,39
3	Subprograms: Function Definition	5	22,29
4	Subprograms: Subroutine Definition	5	10,15,33
5	Input/Output Formats	4	18
6	First Test	-	-
7	FORTRAN on the DEC-10	-	13
8	Libraries and Larger Programs	4	14,30,32,34
9	Debugging and Debug	-	
10	Packages, Plotting etc.	-	
11	Two Dimensional Arrays	12	
12	Use of 2-D arrays in Subprograms	12	
13	No Class	-	-
14	Searching and Sorting	13	37,40
15	Characteristics of Math Functions	6	
16	Final Exam	-	-

TABLE 4

several redundant questions were discarded in favor of others which increased the coverage of those sub-topics which had been inadequately treated.

The chart on the next page shows the distribution of questions by subtopics. Note, that some of the questions appear in more than one of the subtests due to the content of the question.

Finally, the test was prepared using a word processor and the final draft was reviewed by the instructors and by several computer scientists from the National Bureau of Standards who are experts on the FORTRAN programming language. The intent of this double review was to eliminate any content errors and to eliminate any typographical errors. After this review process was completed, the test was printed by the Colorado School of Mines reproduction facility under strict instructions that it was a test and no students were to be present while it was being printed.

DISTRIBUTION OF QUESTIONS BY SUBTEST
SUBTEST NUMBER

Q#	1	2	3	4	5	6	7	8	Q#	USES
1	x								1	1
2		x							2	1
3				x					3	1
4							x		4	1
5	x							x	5	2
6	x							x	6	2
7							x		7	1
8				x					8	1
9	x								9	1
10		x							10	1
11							x		11	1
12				x					12	1
13						x			13	1
14			x						14	1
15		x							15	1
16	x								16	1
17							x		17	1
18			x						18	1
19				x					19	1
20				x					20	1
21				x					21	1
22		x							22	1
23					x				23	1
24				x					24	1
25							x		25	1
26			x					x	26	2
27	x								27	1
28							x		28	1
29					x				29	1
30		x							30	1
31					x				31	1
32			x						32	1
33		x							33	1
34						x	x		34	2
35					x			x	35	2
36							x		36	1
37						x			37	1
38			x		x			x	38	3
39					x			x	39	2
40						x			40	1
TOT	6	6	5	7	6	4	8	6		48

TABLE 5

DATA COLLECTION

One of the more challenging problems to solve was collection of the data. If the testing procedure was to be effective it would be necessary to have all of the subjects take the test on the same day to avoid interactions between subjects on the content of the test. It was neither practical nor possible to require that all of the subjects meet in some central facility so that all could take the test at the same time. What was needed was some natural facility which would capture the maximum number of target subjects under controlled conditions without disrupting the normal course of affairs of the school.

At the suggestion of the Chairman of the Mathematics and Computer Science Department at Colorado School of Mines, it was decided to test all of the students enrolled in all of the sections of Calculus II and Calculus III. The testing was to be done as early as possible in the Fall 1985 Semester. The date chosen was the first Wednesday of classes.

The reasons for this selection of courses were simple. First, all Calculus courses are taught on the same days of the week which would minimize the opportunities for collusion. Second, the majority of the students in these courses would be the target subjects returning for their Sophomore year. Most of the target subjects would be enrolled in Calculus III since they were supposed to have completed Calculus I and II during their Freshman year. The reason for including Calculus II classes in the testing program was to include those students who had taken some remedial mathematics work prior to beginning Calculus and also to include those who had repeated Calculus I.

There were two reasons for the selection of the testing date. First, the test had to be given before the subjects gained significant new programming-related knowledge from the classes in which they were newly enrolled. Second, the test administrators would be the normally assigned instructors for those classes and there was a need to give the classes some opportunity to stabilize in attendance and to provide some opportunity to enlist the assistance of the instructors and to train them in the administration of the test.

The Calculus instructors who administered the test were instructed to read the cover page to their classes after the test had been distributed, and to simply tell the students to take the test. They were instructed to offer no help of any kind to the participants and to deny any knowledge of the content of the test. The test administrators were simply to administer the test as proctors. Several of the test administrators reported that they had been asked for clarification on certain points. All of the test administrators stated that they had not given any assistance of any kind to the students.

In addition to testing the experimental and control group subjects on the specified day, it was decided that two supplementary groups of students were also to be tested. The primary reason for testing these two additional groups of students was to establish an additional dimension of validity for the questions on the test instrument.

First, the assistance of the Mineral Economics Department was solicited to assist in the testing of a large group of newly entering Freshmen. Three large sections of "Introduction To Mineral Economics" with a total enrollment of 215 students were tested. For two of these sections the test administrator was the researcher. The regular professor administered the test to the third section. The purpose for testing this group was to determine whether any of the questions on the test instrument were so easy that someone who had not taken the course could answer them. This information could be used to establish a reasonable lower bound for the test score.

The second supplementary group to be tested on the specified day consisted of those students in three upper division computer science classes normally taught by the researcher. The purpose for testing these upper division students was to establish a test score upper bound, and to determine whether any of the questions might have been too difficult for even experienced subjects to answer.

In the process of entering the results into the computer, a fourth group consisting of those students, who for one reason or another, did not fit the specified criteria for membership in any of the other groups was created. Some examples of individuals who did not fit any of the planned groups are non-Freshmen taking Introduction to Mineral Economics, or upper-classmen taking Calculus II or Calculus III. There were no demographic questions which permitted unambiguous classification of these individuals in any of the other three groups, so they were classified into a composite fourth group as essentially random controls.

CHAPTER 4

RESULTS OF THE STUDY

INTRODUCTION

The purpose of this study was to evaluate the effectiveness of two teaching techniques for teaching introductory computer programming as measured by an achievement test administered at the beginning of the Fall semester following the year in which the instruction took place. The two teaching techniques being evaluated were a traditional lecture-laboratory technique and an experimental multi-disciplinary guided-discovery technique.

This chapter is a report of the results of a comparative analysis of the two techniques. In addition, several demographic factors have been analyzed to determine what relationship, if any, they had to the ability of the students to absorb the material regardless of the teaching technique used.

The order of presentation of the results is essentially the same as previewed at the end of the first chapter. For each primary hypothesis or main effect, first the analysis of the Total Test Score will be discussed followed by a discussion of the analysis for each of the sub-test scores with respect to the main effect. The first results discussed are those for the Teaching Method as a comparison of the traditional versus the experimental methods. These results are followed in order by the analysis of the results by Teacher, Sex of Subject, Computing Courses in High School, Regular Use of a Computer Prior to Entering College, and Ownership of a Personal Computer Prior to Entering College. The final analysis discussed relates to an evaluation of the student's reaction to the teaching methods employed.

For the analyses by Teaching Method and by Teacher an analysis of covariance was used with a Converted College Entrance Examination Score as the covariate to control for pre-treatment aptitude. This was decided when a very highly significant difference was discovered between the mean college entrance examination scores of the control group and the experimental group. This is discussed in Appendix B. Throughout the discussion of results the covariate will be identified as CCHET. A one-way analysis of variance was used to evaluate the Teaching Method Reaction Opinions of the subjects. The demographic factors were also analyzed using a one-way analysis of variance.

HYPOTHESIS TESTING

Analysis of Achievement Scores by Teaching Methods

The purpose of this portion of the study was to determine whether there were any significant differences in mean achievement scores when the subjects were grouped by Teaching Method. A preliminary analysis indicated that there was very highly significant differentiation ($p < 0.001$) between groups on a one-way analysis of variance with Teaching Method as the Independent Variable and Converted College Entrance Examination (CCHET) Score as the Dependent Variable. From this it was decided to control for pre-treatment aptitude by performing analyses of covariance with CCHET as the covariate when analyzing the results by Teaching Method.

Total Test Achievement Score Analysis by Method

The null hypothesis for Total Test Achievement Score analysis by Teaching Method was that there would be no difference in the mean achievement scores of the subjects between the traditional Lecture-Laboratory Group (Control) and the Guided-Discovery Group (Experimental). The Total Test Achievement Scores were analyzed using a one-way analysis of covariance with Teaching Method as the Independent Variable, Total Test

Achievement Score as the Dependent Variable and CCHET as the Control Variable or covariate. The result of this analysis was that there were no significant differences between the mean Total Test Achievement Scores of the two groups. Thus, the null hypothesis cannot be rejected.

Sub-Test Achievement Scores Analyses by Method

A one-way analysis of covariance was performed for each of the eight sub-tests with Teaching Method as the Independent Variable, the appropriate sub-test score as the dependent variable and CCHET as the Control Variable or covariate. For five of the eight sub-tests no significant differences were found. Specifically, there were no significant differences between the mean sub-test scores with Teaching Method as the Independent Variable and CCHET as the covariate for these sub-tests:

1. General FORTRAN Knowledge
2. Knowledge of Functions and Procedures
4. Program Flow Control Statements
6. Miscellaneous Computing Concepts
7. Knowledge of Expressions

For the remaining three sub-tests, significant differences between the means were found.

Input-Output Techniques Score Analysis by Method

The null hypothesis tested for this sub-test was that there would be no significant difference between the mean Input-Output Techniques Sub-Test Achievement Scores between the Lecture-Laboratory (Control) Group and the Guided-Discovery (Experimental) Group. An analysis of covariance with Teaching Method as the Independent Variable, Input-Output Techniques Achievement Sub-Test Scores as the Dependent Variable and CCHET as the covariate revealed that there was a highly significant difference ($p < 0.0001$) between the adjusted mean scores. The adjusted mean Input-Output Techniques Sub-Test Scores of the experimental group and the control group were 2.26 and 1.35 respectively. The adjusted mean scores showed that the experimental group was able to correctly answer nearly one more question than the control group out of the five questions on this sub-test. Tables 4 and 5 on the next page show the results of this analysis in detail.

The analysis of covariance of the Input-Output Techniques Sub-Test Scores by Teaching Method did not support the null hypothesis that there is no difference between the mean scores of the two groups. The

TABLE 6

INPUT-OUTPUT TECHNIQUES SUB-TEST SCORE
GROUP MEANS BY TEACHING METHOD

	Number Subjects	CCHET Score	Adjusted Sub-Test
Experimental Group	72	61.04	2.26
Control Group	88	57.83	1.35
Total	160	59.27	1.76

TABLE 7

INPUT-OUTPUT TECHNIQUES SUB-TEST SCORE
ANALYSIS OF COVARIANCE BY TEACHING METHOD
WITH CCHET AS THE COVARIATE

SOURCE	DF	SUM OF SQUARES	MEAN OF SQUARES	F	SIG-OF-F
Covariate					
CCHET	1	8.56	8.56	8.00	0.01 **
Main Effect					
METHOD	1	30.28	30.28	28.29	0.00 ***
Explained	2	38.85	19.42	18.14	0.00 ***
Residual	157	168.13	1.07		
TOTAL	159	206.98	1.30		

difference of 0.91 between the adjusted mean sub-test scores indicated that the subjects in the experimental group were better able to answer the questions on Input-Output Techniques than were the subjects in the control group. Therefore, the null hypothesis must be rejected.

Ability to Read Programs Score Analysis by Method

The null hypothesis tested for this sub-test analysis was that there would be no significant difference between the mean Ability to Read Programs Sub-Test Achievement Scores between the Lecture-Laboratory (Control) Group and the Guided-Discovery (Experimental) Group. An analysis of covariance with Teaching Method as the Independent Variable, Ability to Read Programs Achievement Sub-Test Scores as the Dependent Variable and CCHET as the covariate revealed that there was a significant difference ($p < 0.05$) between the adjusted mean scores. The adjusted mean Ability to Read Programs Sub-Test Scores of the experimental group and the control group were 2.80 and 2.42 respectively. Tables 6 and 7 on the next page show the results of this analysis.

The analysis of covariance of the Ability to Read Programs Sub-Test Scores by Teaching Method did not

TABLE 8

ABILITY TO READ PROGRAMS SUB-TEST SCORE
GROUP MEANS BY TEACHING METHOD

	Number Subjects	CCHET Score	Adjusted Sub-Test
Experimental Group	72	61.04	2.80
Control Group	88	57.83	2.42
Total	160	59.27	2.59

TABLE 9

ABILITY TO READ PROGRAMS SUB-TEST SCORE
ANALYSIS OF COVARIANCE BY TEACHING METHOD
WITH CCHET AS THE COVARIATE

SOURCE	DF	SUM OF SQUARES	MEAN OF SQUARES	F	SIG-OF-F	
Covariate						
CCHET	1	9.81	9.81	6.96	0.01	**
Main Effect						
METHOD	1	5.57	5.57	3.95	0.05	*
Explained	2	15.38	7.69	5.46	0.01	**
Residual	157	221.21	1.41			
TOTAL	159	236.59	1.49			

support the null hypothesis that there is no difference between the mean scores of the two groups. The difference of 0.38 between the adjusted mean sub-test scores indicated that the subjects in the experimental group were better able to Read Programs and answer the questions on what they had read than were the subjects in the control group. Thus, the null hypothesis may be rejected.

Use of Arrays Sub-Test Score Analysis by Method

This sub-test analysis tested the null hypothesis that there would be no significant difference between the mean scores of the Lecture-Laboratory (Control) Group and the Guided-Discovery (Experimental) Group on the Use of Arrays Sub-Test. An analysis of covariance with Teaching Method as the Independent Variable, Use of Arrays Sub-Test Scores as the Dependent Variable and CCHET as the covariate showed the difference between the adjusted mean scores to be very highly significant ($p < 0.001$). The adjusted mean score for the control group was 1.78 and for the experimental group was 2.40. Tables 8 and 9 on the next page show the results of this analysis.

The difference in adjusted mean scores of 0.62 in favor of the experimental group indicated that the

TABLE 10

USE OF ARRAYS SUB-TEST SCORE
GROUP MEANS BY TEACHING METHOD

	Number Subjects	CCHET Score	Adjusted Sub-Test
Experimental Group	72	61.04	2.40
Control Group	88	57.83	1.78
Total	160	59.27	2.06

TABLE 11

USE OF ARRAYS SUB-TEST SCORE
ANALYSIS OF COVARIANCE BY TEACHING METHOD
WITH CCHET AS THE COVARIATE

SOURCE	DF	SUM OF SQUARES	MEAN OF SQUARES	F	SIG-OF-F
Covariate					
CCHET	1	12.44	12.44	9.27	0.00 ***
Main Effect					
METHOD	1	14.25	14.25	10.63	0.00 ***
Explained	2	26.70	13.35	9.95	0.00 ***
Residual	157	210.68	1.34		
TOTAL	159	237.38	1.49		

experimental group was slightly more knowledgable on the subject covered by the the six questions on this sub-test than was the control group. This was perhaps due to the earlier introduction of the topic to the experimental group in the second week of the term as opposed to the eleventh week for the control group. As a consequence of these findings, the null hypothesis must be rejected.

Discussion of Teaching Method Analyses

The principal null hypotheses to be tested by this study was that there would be no significant difference in the mean achievement test scores for the total test and eight topical sub-tests between the Traditional Lecture-Laboratory (Control) group and the Innovative Guided-Discovery (Experimental) group.

The findings, that there were no significant differences between the adjusted mean scores for the Total Test and five of the eight sub-tests, indicates that there is no great difference which can be measured by a typical achievement test, and that, in general, the null hypotheses cannot be rejected.

For three of the eight sub-tests significant differences were found between the adjusted mean scores of the two groups. Two of these differences were found to be of exactly the same order of magnitude in the same relationships as those found for the same sub-test analyses by Teacher. Therefore, it is probably not possible to determine whether these two differences are related to the teaching method or the teacher.

However, the Ability to Read Programs Sub-Test analysis revealed a significant difference between the means when Teaching Method was the Independent Variable which was not found when Teacher was the Independent Variable. From this it is possible to conclude that the Guided-Discovery, method which depends upon the student to do a great deal of independent reading in the discovery process, better prepares the student to read programs.

Table 10 on the next page is a summary of the results for all nine analyses. The results of the analyses which were not found to be significant were included to show that there is no apparent trend, such that one group consistantly scores higher than the other even though the difference is not significant. Three of the six non-significant differences are in favor of the guided-discovery method and three are in favor of the lecture-laboratory method.

Summary of Results of
Analysis of Covariance
of the Total Test and All Subtests
with TREATMENT as the Independent Variable
and CCHET as the Covariate

INDEPENDENT VARIABLE VALUE	N
EPICS GROUP	72
MA-115 GROUP	88

DEPENDENT VARIABLE	INDEPEND-VAR ADJUSTED MEAN STANDARD-DEV		<----ANCOVA--->		
	EPICS	MA115	F-VAL	F-PROBAB	
Total Test Score	20.82	19.77	2.26	0.14	NS
	4.37	4.63			
Genl FORTRAN Knowledge	3.12	2.94	1.15	0.29	NS
	1.07	0.99			
Functions and Procedures	3.27	3.39	0.30	0.59	NS
	1.42	1.34			
Input-Output Techniques	2.26	1.35	28.28	0.00	***
	1.08	1.00			
Program Flow Control	3.84	3.75	0.17	0.68	NS
	1.22	1.35			
Ability to Read Programs	2.80	2.42	3.95	0.05	**
	1.27	1.14			
Misc Computing Concepts	2.38	2.49	0.46	0.50	NS
	0.82	1.07			
Expressions	3.66	3.88	1.27	0.26	NS
	1.09	1.29			
Use of Arrays	2.40	1.78	10.63	0.00	***
	1.19	1.16			

TABLE 12

Analysis of Student Reaction to Teaching Methods

In order to help determine the acceptability of the experimental program to the students, a Teaching Method Reaction Opinion factor was examined. The null hypothesis to be tested was that there would be no difference in student REACTION to the two teaching methods. In order to measure the REACTION of the students, each group was asked the same question. The subjects were asked,

> "How do you feel that the FORTRAN computer programming portion of the EPICS program compared to the more traditional approach used in MA-115?"

The choices of answers were:

a. Much Better
b. Better
c. About the Same
d. Worse
e. Much Worse
f. I do not know about the other program.

Most of the subjects answered "f" claiming not to know about the other program. For the purposes of the analysis, it was felt that this was better than forcing the students to render an opinion when they really did not have a valid opinion.

Since answers "a" through "e" were essentially on a continuum from "Much Better" to "Much Worse," the letter answers were converted to numeric form for the analysis. This was accomplished by assigning an integer equivalent to the ordinal position of the letter in the alphabet (a=1, b=2, etc.) in place of each letter answer.

When the evaluation was performed only those students who had expressed a real opinion (answers a. through e.) were used. A one-way analysis of variance was used with Teaching Method as the Independent Variable and REACTION as the Dependent Variable. The result of the analysis was that there was no significant difference between the mean REACTION scores of the two groups. The mean score for the 31 students in the EPICS group was 3.00 and for the 43 students in the MA-115 group was 2.93. Clearly, the students who expressed a reaction to the teaching methods were of the opinion that there was no great difference between the two methods.

Analysis of Achievement Scores by Teacher

The purpose of this section of the study was to determine whether there were any significant differences in mean achievement scores when the subjects were grouped by TEACHER. A preliminary one-way analysis of variance with TEACHER of the group as the Independent Variable and Converted College Entrance Examination Scores (CCHET) as the Dependent Variable discovered a highly significant ($p < 0.01$) difference among the mean scores of the four groups (See Appendix B). From this result it was decided to control for pre-treatment aptitude with CCHET by performing analyses of covariance with CCHET as the covariate when analyzing the results of the Total Test and the sub-tests by Teacher.

An important factor which must be considered during the discussion of the analyses of the mean scores by Teacher is that the Experimental Group had only one teacher while there were three teachers for the Control group. This means that there will be no differences within the Experimental Group but there may be differences within the Control Group. Consequently there is probably a strong inter-relationship between the results by Teaching Method and by Teacher.

Total Test Achievement Score Analysis by Teacher

The null hypothesis for Total Test Achievement Score analysis by Teacher was that there would be no differences in mean achievement scores of the subjects among the four Teacher differentiated groups. The Total Test Achievement Scores were analyzed using a one-way analysis of covariance with Teacher as the Independent Variable, Total Test Achievement Score as the Dependent Variable and CCHET as the covariate. The result of this analysis was that there were no significant differences among the mean Total Test Achievement Scores of the four groups. Thus, the null hypothesis cannot be rejected.

Sub-Test Achievement Scores Analyses by Teacher

Eight one-way analyses of covariance were performed, one for each of the eight sub-tests. For each analysis of covariance, Teacher was the Independent Variable, the appropriate sub-test score was the Dependent Variable and CCHET was the covariate. Six of the eight analyses found that there were no significant differences among the mean sub-test scores. Specifically, there were no significant differences among the mean sub-test scores of the four groups with

Teacher as the Independent Variable and CCHET as the covariate for the following sub-tests:

1. General FORTRAN Knowledge
2. Functions and Procedures
4. Program Flow Control
5. Ability to Read Programs
6. Miscellaneous Computing Concepts
7. Expressions

The analyses for the remaining two sub-tests found significant differences among the mean scores which are described in detail in the following sections.

Input-Output Techniques Score Analysis by Teacher

The null hypothesis which was tested in the analysis of the results for the Input-Output Techniques Sub-Test was that there would be no significant differences in the mean scores among the four groups differentiated by Teacher. An analysis of covariance was performed with Teacher as the Independent Variable, Input-Output Techniques Sub-Test Achievement Scores as the Dependent Variable and CCHET as the covariate. The analysis of covariance showed that there was a very highly significant difference ($p < 0.0001$) among the adjusted mean scores. Tables 11 and 12 on the next page show the results of this analysis.

TABLE 13

INPUT-OUTPUT TECHNIQUES SUB-TEST SCORE
GROUP MEANS BY TEACHER

	Number Subjects	CCHET Score	Adjusted Sub-Test
Experimental Group:			
Teacher-1	72	61.04	2.26
Control Group:			
Teacher-2	32	56.39	1.53
Teacher-3	26	58.79	1.36
Teacher-4	29	58.48	1.19
Total	160	59.27	1.76

TABLE 14

INPUT-OUTPUT TECHNIQUES SUB-TEST SCORE
ANALYSIS OF COVARIANCE BY TEACHER
WITH CCHET AS THE COVARIATE

SOURCE	DF	SUM OF SQUARES	MEAN OF SQUARES	F	SIG-OF-F	
Covariate						
CCHET	1	8.60	8.60	7.97	0.01	**
Main Effect						
TEACHER	3	31.53	10.51	9.73	0.00	***
Explained	4	40.13	10.03	9.29	0.00	***
Residual	154	166.26	1.08			
TOTAL	158	206.39	1.31			

The adjusted mean Input-Output Techniques Sub-Test Achievement Scores are 2.26 for the teacher of the experimental group and 1.53, 1.36, and 1.19 respectively for the three teachers of the control group. A review of the means analysis shows that the mean score of the group for the teacher of the experimental group to be much higher than the others all of which were below the grand mean of 1.77. The maximum difference of 1.05 between Teacher-1 and Teacher-4 indicates that the best group was able to answer more than one more question correctly than the worst group. A Newman-Keul analysis using the harmonic mean of the group "n's" as the working "n" found that the only significant differences between the adjusted mean scores of the four groups was between Teacher-1 and all of the other teachers. In any event, the very high significance of the differences indicates that the null hypothesis must be rejected for this sub-test.

Use of Arrays Sub-Test Score Analysis by Teacher

This sub-test analysis tested the null hypothesis that there would be no significant differences among the mean scores of the four Teacher differentiated groups on the Use of Arrays Sub-Test. An analysis of covariance with Teacher as the Independent Variable, Use of Arrays Sub-Test Scores as the Dependent Variable and CCHET as the covariate showed the difference to be highly significant ($p < 0.01$). The adjusted mean score for Teacher-1, the teacher of the Experimental Group was 2.40. The adjusted mean scores for the teachers of the Control Group were 1.78, 2.05, and 1.58 for Teachers 2, 3, and 4 respectively. The maximum difference was between Teacher-1 and Teacher-4 and was 0.82. The minimum difference was between Teacher-2 and Teacher-4 and was 0.20. Tables 13 and 14 on the next page show the results of this analysis. A Newman-Keul analysis using the harmonic mean of the group "n's" as the working "n" found significant differences between Teacher-1 and Teachers-3 and Teacher-1 and Teacher-4. No other significant differences were found between teachers.

Considering the results of the analysis of covariance, the null hypothesis must be rejected for this sub-test.

Discussion of Teacher Analyses

A secondary but very important set of null hypotheses to be tested by this study was that there would be no significant differences in the mean

TABLE 15

USE OF ARRAYS SUB-TEST SCORE
GROUP MEANS BY TEACHER

	Number Subjects	CCHET Score	Adjusted Sub-Test
Experimental Group:			
Teacher-1	72	61.04	2.40
Control Group:			
Teacher-2	32	56.39	1.78
Teacher-3	26	58.79	2.05
Teacher-4	29	58.48	1.58
Total	160	59.27	2.07

TABLE 16

USE OF ARRAYS SUB-TEST SCORE
ANALYSIS OF COVARIANCE BY TEACHER
WITH CCHET AS THE COVARIATE

SOURCE	DF	SUM OF SQUARES	MEAN OF SQUARES	F	SIG-OF-F
Covariate					
CCHET	1	12.51	12.51	9.31	0.00 ***
Main Effect					
TEACHER	3	16.85	5.62	4.18	0.01 **
Explained	4	29.36	7.34	5.46	0.00 ***
Residual	154	206.88	1.34		
TOTAL	158	236.24	1.50		

achievement test scores for the Total Test and the eight topical sub-tests among the four groups differentiated by teacher.

The findings, that there were no significant differences among the adjusted mean scores for the Total Test and six of the eight sub-tests, indicate that there was no great difference among the teachers which can be measured by a typical achievement test, and that, in general, the null hypotheses cannot be rejected.

For two of the eight sub-tests significant differences were found among the adjusted mean scores of the four groups. Newman-Keul analyses of these differences revealed that the only significant differences were between Teacher-1 and the other teachers. Furthermore, these differences were found to have exactly the same relationship and level of significance as was found for the same sub-test analyses with Teaching Method as the Independent Variable. Therefore, it is probably not possible to distinguish whether the method or the teacher is the source of the variation.

Table 15, on the next page is a summary of the results for all nine analyses. The results for the

TABLE 17

Summary of Results of Analyses of Variance
Of the Total Test and All Subtests
With TEACHER as the Independent Variable

INDEPENDENT VARIABLE VALUE	N
TEACHER-1 (EPICS)	72
TEACHER-2 (MA-115)	32
TEACHER-3 (MA-115)	26
TEACHER-4 (MA-115)	29

DEPENDENT VARIABLE	INDEPENDENT-VARIABLE ADJUSTED MEAN STANDARD-DEVIATION				<---ANCOVA--->		
	T-1	T-2	T-3	T-4	F-VAL	F-PROBAB	
Total Test	20.80	20.26	20.20	18.85	1.42	0.24	NS
	4.37	4.85	4.96	4.17			
Genl FORTRAN	3.12	2.70	3.17	2.99	1.42	0.24	NS
	1.07	1.16	0.78	0.94			
Funcs & Procs	3.27	3.84	3.36	2.89	2.63	0.052	NS
	1.42	1.20	1.06	1.57			
Input-Output	2.26	1.53	1.36	1.19	9.73	0.00	***
	1.08	1.14	1.02	0.85			
Prog Flow Ctl	3.84	3.83	3.91	3.63	0.27	0.85	NS
	1.22	1.35	1.34	1.32			
Read Programs	2.80	2.46	2.47	2.26	1.63	0.19	NS
	1.27	1.18	1.21	1.02			
Misc Concepts	2.38	2.69	2.18	2.56	1.73	0.16	NS
	0.82	1.16	1.08	0.95			
Expressions	3.66	3.82	4.02	3.83	0.64	0.59	NS
	1.09	1.20	1.26	1.42			
Use of Arrays	2.40	1.78	2.05	1.58	4.18	0.007	***
	1.19	1.06	1.31	1.12			

tests which were not found to be significant were included in this summary to show that there is no apparent trend, such that one teacher out of the four consistantly scores higher than the others even though the difference may not be significant. Of the seven non-significant results, two of the teachers each had two highest scores, one teacher had the other three highest scores, and one teacher had none.

Analysis of Achievement Scores by Sex of Subject

As part of the analysis of each demographic factor associated with the subjects of the study, ten analyses of variance were performed. First, in order to determine whether it might be necessary to control for pre-treatment aptitude an analysis of variance was done with SEX as the Independent Variable and Converted College Entrance Examination Scores (CCHET) as the Dependent Variable. The distribution of subjects by sex was 32 females and 128 males. No significant difference was found between the mean CCHET scores of the Males and the Females. Therefore, it was determined that a one-way analysis of variance would be the appropriate tool for analyzing the achievement scores by Sex of subject.

Total Test Achievement Score Analysis by Sex of Subject

The null hypothesis for Total Test Achievement Score analysis by Sex of Subject was that there would be no difference in the mean scores of the subjects when they were differentiated by sex. The Total Test Achievement Scores were analyzed using a one-way analysis of variance with Sex as the Independent Variable and Total Test Achievement Score as the Dependent Variable. The result of this analysis was that there were no significant differences between the mean Total Test Achievement Scores when the subjects were differentiated by sex. Therefore, the null hypothesis cannot be rejected.

Sub-Test Achievement Scores Analysis by Sex of Subject

The null hypothesis to be tested for each of the eight sub-tests with respect to sex of the subject was that there would be no significant differences between the mean sub-test scores of the subjects when differentiated by Sex. Eight one-way analyses of variance were done, each with Sex of the Subject as the Independent Variable and one of the eight sub-test scores as the Dependent Variable. These analyses found no significant differences between the mean sub-test scores of groups differentiated by Sex of the Subject. Therefore, these null hypotheses cannot be rejected.

Discussion of Sex of Subject Analyses

No significant differences were found between the mean Total Test and Sub-Test scores of the subjects when they were grouped by sex. This implies that the sex of the student is not a factor affecting success in introductory computer programming courses at the Colorado School of Mines.

Analysis of Scores by High School Computing Course

The purpose for this demographic analysis was to determine whether having taken a computing course would affect achievement in a college level computing course. The question asked of the participants was: "Did You Take Any Computing Courses in High School?" The only answers permitted were "Yes" and "No." The responses were distributed Yes(122) and No(38).

Prior to performing the analyses of the achievement test scores, a one-way analysis of variance was done with High School Computing Course (HSCC) as the Independent Variable and the Converted College Entrance Examination Score as the Dependent Variable. No significant difference was found between the mean CCHET scores of those who took a computing course in high school and those who did not. Therefore, it was decided to use a simple one-way analysis of variance to analyze the achievement scores of the subjects.

Total Test Score Analysis by HS Computing Course

The null hypothesis for evaluation of Total Test Achievement Score by High School Computing Course was that there would be no significant difference between the scores of those who had taken a high school computing course and those who had not taken such a course. The Total Test Achievement Scores were analyzed using a one-way analysis of variance with High School Computing Course as the Independent Variable and Total Test Achievement Score as the Dependent Variable. The result of this analysis was that there was a very highly significant difference ($p < 0.001$) between the mean scores of the two groups. The mean score for the HSCC-YES group was 20.96 and the mean score for the HSCC-NO group was 17.92. Tables 16 and 17 on the following page show the results of this analysis in detail.

The difference between the mean scores of 3.04 in favor of the HSCC-YES group indicated that the HSCC-YES group was somewhat more knowledgable on the Total Test than was the HSCC-NO group. The difference was slightly more than two-thirds of a standard deviation either way. As a consequence of these findings, the null hypothesis must be rejected.

The magnitude and direction of the difference in favor of those who had taken a computing course in high school indicates that taking such a course is indeed an excellent preparation for a college computing course. This tends to support Ramberg's findings (1986).

TABLE 18

TOTAL TEST SCORE
GROUP MEANS BY HIGH SCHOOL COMPUTING COURSE

Group	Number Subjects	Mean Scores
HSCC-YES	122	20.96
HSCC-NO	38	17.92
Total	160	20.24

TABLE 19

TOTAL TEST SCORES
ANALYSIS OF VARIANCE BY
HIGH SCHOOL COMPUTING COURSE

SOURCE	DF	SUM OF SQUARES	MEAN OF SQUARES	F	SIG-OF-F
Main Effect					
HSCC	1	267.41	267.41	13.70	0.00 ***
Residual	158	3083.56	19.52		
TOTAL	159	3350.97	21.08		

Sub-Test Score Analysis by HS Computing Course

A one-way analysis of variance was performed for each of the eight sub-tests with High School Computing Course as the Independent Variable and the appropriate sub-test score as the Dependent Variable. For four of the eight sub-tests no significant differences were found. Specifically, there were no significant differences between the mean sub-test scores with High School Computing Course as the Independent variable for these sub-tests:

1. General FORTRAN Knowledge
2. Knowledge of Functions and Procedures
3. Input-Output Techniques
8. Use of Arrays

For the remaining four sub-tests, significant differences between the mean scores were found.

Program Flow Control Subtest Score Analysis by HSCC

The null hypothesis tested for this sub-test was that there would be no significant difference between the mean Program Flow Control Sub-Test Achievement Scores of the subjects when they were grouped by whether they had taken a computing course in high school. A one-way analysis of variance with HSCC as the Independent Variable and Program Flow Control

Concepts Sub-Test Achievement Scores as the Dependent Variable revealed that there was a very highly significant difference ($p < 0.0001$) between the mean scores of the two groups. The mean Program Flow Control Concepts Sub-Test Score of the subjects who had taken a computing course in high school was 3.98, while the mean score for those who had not taken such a course was 3.16. Tables 18 and 19 on the next page show the details of the results of this analysis.

The mean scores showed that the subjects who answered YES to the question of whether they had taken a high school computing course were able to correctly answer almost one more (0.82) question about Program Flow Control than the subjects who answered NO. Program Flow Control is one of the more difficult concepts to teach. Therefore, previous experience with this topic would tend to make it easier to understand in a subsequent course, which would tend to account for the higher scores of those who took a computing course prior to taking either EPICS or MA-115.

This analysis did not support the null hypothesis that there was no difference between the mean scores of the two groups for this sub-test. Therefore, the null hypothesis must be rejected.

TABLE 20

PROGRAM FLOW CONTROL SUB-TEST SCORE
GROUP MEANS BY HIGH SCHOOL COMPUTING COURSE

Group	Number Subjects	Mean Scores
HSCC-YES	122	3.98
HSCC-NO	38	3.16
Total	160	3.78

TABLE 21

PROGRAM FLOW CONTROL SUB-TEST SCORE
ANALYSIS OF VARIANCE BY
HIGH SCHOOL COMPUTING COURSE

SOURCE	DF	SUM OF SQUARES	MEAN OF SQUARES	F	SIG-OF-F
Main Effect					
HSCC	1	19.76	19.76	12.64	0.00 ***
Residual	158	247.02	1.56		
TOTAL	159	266.78	1.68		

Ability to Read Programs Subtest Score Analysis by HSCC

The null hypothesis tested for this sub-test was that there would be no significant difference between the mean Ability to Read Programs Sub-Test Achievement Scores of the subjects when they were grouped by whether they had taken a computing course in high school. A one-way analysis of variance with HSCC as the Independent Variable and Ability to Read Programs Sub-Test Achievement Scores as the Dependent Variable revealed that there was a very highly significant difference ($p < 0.0001$) between the mean scores of the two groups. The mean Ability to Read Programs Sub-Test Score of the subjects who had taken a computing course in high school was 2.77, while the mean score for those who had not taken such a course was 2.03. Tables 20 and 21 on the next page show the details of the results of this analysis.

The difference between the mean scores of the two groups of 0.74 showed that the subjects who answered YES to the question of whether they had taken a high school computing course were able to correctly answer nearly one more question involving the reading of programs than the subjects who answered NO. Skill in reading programs is typically gained by reading

TABLE 22

ABILITY TO READ PROGRAMS SUB-TEST SCORE
GROUP MEANS BY HIGH SCHOOL COMPUTING COURSE

Group	Number Subjects	Mean Scores
HSCC-YES	122	2.77
HSCC-NO	38	2.03
Total	160	2.59

TABLE 23

ABILITY TO READ PROGRAMS SUB-TEST SCORE
ANALYSIS OF VARIANCE BY
HIGH SCHOOL COMPUTING COURSE

SOURCE	DF	SUM OF SQUARES	MEAN OF SQUARES	F	SIG-OF-F
Main Effect					
HSCC	1	16.05	16.05	11.50	0.00 ***
Residual	158	220.55	1.40		
TOTAL	159	236.59	1.49		

programs, and almost all computing courses involve some reading of programs. Therefore, taking a computing course in high school would tend to enhance the ability to read programs and, hence, to make higher scores on questions relating to reading programs.

The null hypothesis that there would be no difference between the mean scores of the two groups on the Ability to Read Programs is not supported by the results of this analysis. Therefore, the null hypothesis must be rejected.

Miscellaneous Computing Concepts Score Analysis by HSCC

The null hypothesis which was tested in the analysis of the results for the Miscellaneous Computing Concepts Sub-Test was that there would be no significant difference between the mean scores of the two groups differentiated by whether they had taken a high school computing course. A one-way analysis of variance with HSCC as the Independent Variable and Miscellaneous Computing Concepts Sub-Test Achievement Scores as the Dependent Variable revealed that there was a highly significant difference ($p < 0.01$) between the mean scores of the two groups. Tables 22 and 23 on the next page show the details of the results of this analysis. The mean score of the subjects who had taken

TABLE 24

MISCELLANEOUS COMPUTING CONCEPTS SUB-TEST SCORE GROUP MEANS BY HIGH SCHOOL COMPUTING COURSE

Group	Number Subjects	Mean Scores
HSCC-YES	122	2.55
HSCC-NO	38	2.08
Total	160	2.44

TABLE 25

MISCELLANEOUS COMPUTING CONCEPTS SUB-TEST SCORE ANALYSIS OF VARIANCE BY HIGH SCHOOL COMPUTING COURSE

SOURCE	DF	SUM OF SQUARES	MEAN OF SQUARES	F	SIG-OF-F	
Main Effect						
HSCC	1	6.41	6.41	7.18	0.01	**
Residual	158	140.97	0.89			
TOTAL	159	147.38	0.97			

a computing course in high school was 2.55, and the mean score for those who had not was 2.08.

The difference between the mean scores of 0.47 in favor of the subjects who had taken a computing course in high school indicates that they were slightly better able to answer general computing questions than those who did not take such a course. Typically, knowledge of general concepts on any subject is cummulative so that greater exposure implies greater familiarity with the subject. The results of this analysis would seem to support this position and indicate that the null hypothesis must be rejected for this sub-test.

Use of Expressions Sub-Test Score Analysis by HSCC

The null hypothesis tested in the evaluation of the results for the Use of Expressions Sub-Test was that there would be no significant difference between the mean scores of the subjects when they were grouped by whether they had taken a high school computing course. A one-way analysis of variance with HSCC as the Independent Variable and Use of Expression Sub-Test Scores as the Dependent Variable revealed a significant difference ($p < 0.05$) between the mean scores of the two groups. Tables 24 and 25 on the next page show the

TABLE 26

USE OF EXPRESSIONS SUB-TEST SCORE
GROUP MEANS BY HIGH SCHOOL COMPUTING COURSE

Group	Number Subjects	Mean Scores
HSCC-YES	122	3.89
HSCC-NO	38	3.42
Total	160	3.78

TABLE 27

USE OF EXPRESSIONS SUB-TEST SCORE
ANALYSIS OF VARIANCE BY
HIGH SCHOOL COMPUTING COURSE

SOURCE	DF	SUM OF SQUARES	MEAN OF SQUARES	F	SIG-OF-F	
Main Effect						
HSCC	1	6.24	6.24	4.45	0.04	*
Residual	158	221.66	1.40			
TOTAL	159	227.90	1.43			

details of the results of this analysis. The mean Use of Expressions Score of the subjects who had taken a computing course in high school was 3.89, and the mean score for those who had not was 3.42.

The difference between the mean scores of 0.47 in favor of the subjects who had taken a computing course in high school indicates that they were slightly better able to answer questions about the Use of Expressions than the subject who did not take such a course. Since most computing courses involving programming must make some reference to use of expressions, it seems likely that those who had taken a previous course in computing would be slightly more familiar with the topic than those who had not. The significant difference between the mean sores on the Use of Expressions Sub-Test, found by the one-way analysis of variance by HSCC, indicates that the null hypothesis must be rejected for this sub-test.

Discussion of High School Computing Course Analyses

Nine null hypotheses were tested in the evaluation of whether or not taking a high school computing course was good preparation for taking a college level computing course. The primary hypothesis tested here was that there would be no significant differences between the mean total scores of the subjects when they were separated by whether they had taken a computing course in high school. Eight secondary hypotheses were tested to determine whether there would be any significant differences between the mean scores in eight topical areas with the same differentiation.

The main reason for performing the primary test was to determine whether a high school computing course was a good preparation for a college computing course. The secondary tests were performed to determine in what topical areas the differences, if any, would be most likely to appear.

Very highly significant differences were found between the means for the Total Test Score, which indicates that Ramberg (1986) is probably correct in his assertion. For two of the eight sub-tests, Program Flow Control and Ability to Read Programs, very highly significant differences ($p < 0.001$) were found between the means of the two groups. For the Miscellaneous Computing Concepts Sub-Test, highly significant differences ($p < 0.01$) were found and significant differences between the means were found for the Use of Expressions Sub-Test. All four of these results tend to support Ramberg's assertion since they represent computer science concepts which are likely to be common to all computing courses.

Table 24 on the next page is a summary of the results of all nine of the analyses by High School Computing Course. The results for the sub-test analyses for which no significant differences were found have been included to show that there is a definite trend in favor of taking a computing course in high school. Note that all nine results are in favor of the group which took a computing course in high school.

Summary of Results of
Analyses of Variance
Of the Total Test and All Subtests With
HIGH SCHOOL COMPUTING COURSE
as the Independent Variable

INDEPENDENT VARIABLE VALUE	N
HSCC-YES	122
HSCC-NO	38

DEPENDENT VARIABLE	INDEPEND-VAR MEAN SCORE STANDARD-DEV		<----ANOVA---->		
	YES	NO	F-VAL	F-PROBAB	
Total Test Score	20.96	17.92	13.70	0.00	***
	4.53	4.02			
Genl FORTRAN Knowledge	3.03	2.97	0.09	0.76	NS
	1.03	1.08			
Functions and Procedures	3.44	3.03	2.71	0.10	NS
	1.40	1.24			
Input-Output Techniques	1.82	1.58	1.29	0.26	NS
	1.19	0.98			
Program Flow Control	3.98	3.16	12.64	0.00	***
	1.21	1.39			
Ability to Read Programs	2.77	2.03	11.50	0.00	***
	1.19	1.15			
Misc Computing Concepts	2.55	2.08	7.18	0.01	**
	0.94	0.97			
Use of Expressions	3.89	3.42	4.45	0.04	*
	1.11	1.21			
Use of Arrays	2.11	1.89	0.94	0.33	NS
	1.22	1.23			

TABLE 28

Analysis of Score by Computer Use Prior to College

The purpose of this demographic analysis was to determine whether regular prior exposure to computers was a good preparation for a computing course. The question asked of the participants was: **"Did You Have Access to and Make Use of a Computer Prior to Entering Colorado School of Mines?"** The only responses permitted were "Yes" and "No." The responses were Yes(73) and No(87).

Before the achievement test score analyses were performed, a one-way analysis of variance was done with Regular Use of Computer Prior to College (RUCPC) as the Independent Variable and the Converted College Entrance Examination Score as the Dependent Variable. No significant difference was found between the mean CCHET scores of those who responded YES and those who responded NO. Therefore, it was decided to use a simple one-way analysis of variance to analyze the achievement scores of the subjects.

Total Test Score Analysis by Prior Use of Computer

The null hypothesis for evaluation of Total Test Achievement Score by Prior Use of Computer was that there would be no significant difference between the scores of those who answered RUCPC-YES and those who answered RUCPC-NO. The Total Test Achievement Scores were analyzed using a one-way analysis of variance with Regular Use of a Computer Prior to College (RUCPC) as the Independent Variable and Total Test Achievement Score as the Dependent Variable. This analysis revealed that there was a highly significant difference ($p < 0.01$) between the mean scores of the two groups. The mean scores were 21.30 and 19.34 for the RUCPC-YES and the RUCPC-NO groups respectively. Tables 27 and 28 on the following page show the results of this analysis in detail.

The difference between the mean scores was 1.96 in favor of those who had made Regular Use of a Computer Prior to College. This indicated that the RUCPC-YES group was somewhat more knowledgable on the Total Test than was the RUCPC-NO group. The difference represents approximately 10-percent of the grand mean and clearly indicates that prior exposure to computers through Regulars Use of a Computer Prior to College enhances the likelihood of success in a college computing course. As a consequence of these findings, the null hypothesis must be rejected. This tends to support Ramberg's findings (1986).

TABLE 29

TOTAL TEST SCORE GROUP MEANS BY
REGULAR USE OF A COMPUTER PRIOR TO COLLEGE

Group	Number Subjects	Mean Scores
RUCPC-YES	73	21.30
RUCPC-NO	87	19.34
Total	160	20.24

TABLE 30

TOTAL TEST SCORES
ANALYSIS OF VARIANCE BY
REGULAR USE OF A COMPUTER PRIOR TO COLLEGE

SOURCE	DF	SUM OF SQUARES	MEAN OF SQUARES	F	SIG-OF-F	
Main Effect						
RUCPC	1	151.95	151.95	7.50	0.01	**
Residual	158	3199.03	20.25			
TOTAL	159	3350.97	21.08			

Sub-Test Score Analysis by Prior Use of Computer

A one-way analysis of variance was performed for each of the eight sub-tests with Regular Use of a Computer Prior to College as the Independent Variable and the appropriate sub-test score as the Dependent Variable. Four of the analyses found no significant differences. Specifically, there were no significant differences between the mean sub-test scores with Regular Use of a Computer Prior to College as the Independent variable for the following sub-tests:

1. General FORTRAN Knowledge
3. Input-Output Techniques
5. Ability to Read Programs
8. Use of Arrays

Significant differences between the mean scores were found for the remaining four sub-tests.

Functions and Procedures Sub-Test Analysis by RUCPC

This analysis tested the null hypothesis that there would be no significant differences between the mean Functions and Procedures Sub-Test Scores of those who had made Regular Use of a Computer Prior to College and those who had not. A one-way analysis of variance with RUCPC as the Independent Variable and Functions and Procedures Sub-Test Scores as the Dependent

Variable found that there was a significant difference ($p < 0.05$) between the two mean scores. The RUCPC-YES group mean score was 3.59, and the RUCPC-NO group mean score was 3.14. Tables 29 and 30 on the next page show the results of this analysis.

The results show that the subjects who made Regular Use of a Computer Prior to College scored 0.45 higher than those who did not on this sub-test. The difference is not great, but there appears to be a definite benefit in the understanding of Functions and Procedures from prior use of computers.

This analysis did not support the null hypothesis that there was no difference between the mean scores of the two groups for this sub-test. Therefore, the null hypothesis must be rejected.

Program Flow Control Sub-Test Analysis by RUCPC

The null hypothesis tested for this sub-test was that there would be no significant difference between the mean Program Flow Control Sub-Test Achievement Scores of the subjects when they are grouped by whether they had made Regular Use of a Computer Prior to College (RUCPC). A one-way analysis of variance with RUCPC as the Independent Variable and Program Flow

TABLE 31

FUNCTIONS AND PROCEDURES SUB-TEST SCORE
GROUP MEANS BY
REGULAR USE OF A COMPUTER PRIOR TO COLLEGE

Group	Number Subjects	Mean Scores
RUCPC-YES	73	3.59
RUCPC-NO	87	3.14
Total	160	3.35

TABLE 32

FUNCTIONS AND PROCEDURES SUB-TEST SCORE
ANALYSIS OF VARIANCE BY
REGULAR USE OF A COMPUTER PRIOR TO COLLEGE

SOURCE	DF	SUM OF SQUARES	MEAN OF SQUARES	F	SIG-OF-F	
Main Effect						
RUCPC	1	8.08	8.08	4.40	0.04	*
Residual	158	290.02	1.84			
TOTAL	159	298.09	1.87			

Control Sub-Test Score as the Dependent Variable revealed that there was a significant difference between the mean scores of the two groups. The mean Program Flow Control Sub-Test Score of the RUCPC-YES group was 4.03, while the mean of the RUCPC-NO group was 3.59. Tables 31 and 32 on the following page show the results of this analysis in detail.

The mean scores showed that the subjects who answered YES to the question of whether they had made Regular Use of a Computer Prior to College were able to score 0.44 higher on questions relating to Program Flow Control than those who answered NO. This may be due to the need for regular and logical thinking associated with effective use of a computer. Therefore, it would seem that prior use of a computer is of some use in preparing for a college level computing course.

These findings do not support the null hypothesis that there would be no significant difference between the mean scores. As a result, the null hypothesis must be rejected.

Misc. Computing Concepts Sub-Test Analysis by RUCPC

The null hypothesis which was tested in the analysis of the results for the Miscellaneous Computing Concepts Sub-Test was that there would be no

TABLE 33

PROGRAM FLOW CONTROL SUB-TEST SCORE
GROUP MEANS BY
REGULAR USE OF A COMPUTER PRIOR TO COLLEGE

Group	Number Subjects	Mean Scores
RUCPC-YES	73	4.03
RUCPC-NO	87	3.59
Total	160	3.79

TABLE 34

PROGRAM FLOW CONTROL SUB-TEST SCORE
ANALYSIS OF VARIANCE BY
REGULAR USE OF A COMPUTER PRIOR TO COLLEGE

SOURCE	DF	SUM OF SQUARES	MEAN OF SQUARES	F	SIG-OF-F	
Main Effect						
RUCPC	1	7.73	7.73	4.71	0.03	*
Residual	158	259.05	1.64			
TOTAL	159	266.78	1.68			

significant difference between the mean scores of the two groups differentiated by whether they had made Regular Use of a Computer Prior to College. A one-way analysis of variance with RUCPC as the Independent Variable and Miscellaneous Computing Concepts Sub-test Score as the Dependent Variable found a very highly significant difference ($p < 0.0001$) between the mean scores of the two groups. Tables 33 and 34 on the next page show the results of this analysis in detail. The mean score of the subjects who answered RUCPC-YES was 2.75, and for those who answered RUCPC-NO it was 2.17.

The difference between the mean scores of 0.58 in favor of the subjects who answered RUCPC-YES indicates that they were somewhat better able to answer questions relating to Miscellaneous Computing Concepts than those who had answered RUCPC-NO. This difference is nearly 25-percent of the grand mean for this sub-test.

All computer usage involves the concepts of this sub-test to some extent. Thus, the more regularly the computer is used, the more likely the individual is to do well on a test of these concepts.

The finding of a very highly significant difference between the means for this sub-test means the null hypothesis must be rejected.

TABLE 35

MISCELLANEOUS COMPUTING CONCEPTS SUB-TEST SCORE GROUP MEANS BY REGULAR USE OF A COMPUTER PRIOR TO COLLEGE

Group	Number Subjects	Mean Scores
RUCPC-YES	73	2.75
RUCPC-NO	87	2.17
Total	160	2.43

TABLE 36

MISCELLANEOUS COMPUTING CONCEPTS SUB-TEST SCORE ANALYSIS OF VARIANCE BY REGULAR USE OF A COMPUTER PRIOR TO COLLEGE

SOURCE	DF	SUM OF SQUARES	MEAN OF SQUARES	F	SIG-OF-F
Main Effect					
RUCPC	1	13.40	13.40	15.80	0.00 ***
Residual	158	133.98	0.85		
TOTAL	159	147.38	0.93		

Use of Expressions Sub-Test Analysis by RUCPC

The null hypothesis tested in the evaluation of the results of the Use of Expressions Sub-Test was that there would be no significant difference between the mean scores of the two groups differentiated by whether they had made Regular Use of a Computer Prior to College. A one-way analysis of variance with RUCPC as the Independent Variable and Use of Expressions Sub-test Score as the Dependent Variable found a very highly significant difference ($p < 0.0001$) between the mean scores of the two groups. The mean score of the subjects who had made Regular Use of a Computer Prior to College was 4.15. For those who responded that they had not made Regular Use of a Computer Prior to College, the mean score was 3.45. Tables 35 and 36 on the next page show the results of this analysis in detail.

The subjects who answered RUCPC-YES scored 0.70 higher than those who answered RUCPC-NO which indicates that they were better prepared in the Use of Expressions.

To some extent, all computer usage involves the use of expressions as this is, normally, the primary means of actually computing results. Therefore, it

TABLE 37

USE OF EXPRESSIONS SUB-TEST SCORE
GROUP MEANS BY
REGULAR USE OF A COMPUTER PRIOR TO COLLEGE

Group	Number Subjects	Mean Scores
RUCPC-YES	73	4.15
RUCPC-NO	87	3.45
Total	160	3.76

TABLE 38

USE OF EXPRESSIONS SUB-TEST SCORE
ANALYSIS OF VARIANCE BY
REGULAR USE OF A COMPUTER PRIOR TO COLLEGE

SOURCE	DF	SUM OF SQUARES	MEAN OF SQUARES	F	SIG-OF-F
Main Effect					
RUCPC	1	18.95	18.95	14.33	0.00 ***
Residual	158	208.95	1.32		
TOTAL	159	227.90	1.43		

would seem that regular use of a computer prior to taking a computing course would be good preparation for this topic. The knowledge gained is cummulative so that the more experience gained the more likely the individual is to do well on a test on the subject.

The finding of a very highly significant difference between the means for this sub-test means the null hypothesis must be rejected.

Discussion of Regular Use of Computers Prior to College

The evaluation of whether or not Regular Use of a Computer Prior to College had an affect on achievement in an introductory computer programming course involved the testing of nine null hypotheses. The primary null hypothesis tested was that there would be no significant difference between the mean Total Test scores of the subjects when they were differentiated by whether or not they had Made Regular Use of a Computer Prior to College. In addition to testing the primary hypothesis, eight secondary hypotheses were tested to determine whether there were any significant differences between the mean scores with the same differentiation.

The main reason for examining this factor was to determine whether having access to and making regular use of a computer prior to entering college would be good preparation for taking a college level computing course. The main reason for performing the primary hypothesis test was to answer this question. The secondary analyses were performed to determine in which topical areas the differences, if any, would be most likely to occur.

Highly significant differences ($p < 0.01$) were found between the means of the Total Test Scores with RUCPC as the Independent Variable which means that Ramberg (1986) is probably correct. Very highly significant differences ($p < 0.0001$) were found between the means for two of the sub-tests, Miscellaneous Computing Concepts and Use of Expressions. Significant differences ($p < 0.05$) were found between the means for two other sub-tests, Functions and Procedures and Program Flow Control. These four results all tend to support Ramberg's assertion since they are typical of computing oriented activities.

Table 37 on the next page is a summary of the results of all nine of the analyses with Regular Use of a Computer Prior to College as the Independent Variable. The results with non-significant differences have been included for comparison.

Summary Of Results Of The Analyses of Variance
Of the Total Test and All Subtests With
REGULAR USE OF A COMPUTER PRIOR TO COLLEGE
as the Independent Variable

INDEPENDENT VARIABLE VALUE	N
RUCPC-YES	73
RUCPC-NO	87

DEPENDENT VARIABLE	INDEPEND-VAR MEAN SCORE STANDARD-DEV		<-----ANOVA----> F-VAL	F-PROBAB	
	YES	NO			
Total Test Score	21.30	19.34	7.50	0.01	**
	4.77	4.26			
Genl FORTRAN Knowledge	2.95	3.08	0.67	0.41	NS
	1.03	1.05			
Functions and Procedures	3.59	3.14	4.40	0.04	*
	1.39	1.32			
Input-Output Techniques	1.81	1.72	0.21	0.64	NS
	1.18	1.11			
Program Flow Control	4.03	3.59	4.71	0.03	*
	1.19	1.35			
Ability to Read Programs	2.56	2.62	0.09	0.76	NS
	1.19	1.25			
Misc Computing Concepts	2.75	2.17	15.80	0.00	***
	1.00	0.85			
Use of Expressions	4.15	3.45	14.33	0.00	***
	1.25	1.05			
Use of Arrays	1.93	2.17	1.55	0.22	NS
	1.11	1.30			

TABLE 39

Analysis of Scores by Personal Computer Before College

The purpose for this demographic analysis was to determine whether ownership of a personal computer before entering college would have any influence on performance in an introductory college computing course. The question asked of the participants was, **"Do you or your family own a personal or home computer for which you wrote programs prior to taking your first computing course at Colorado School of Mines?"** The only answered allowed were "Yes" and "No." The responses were distributed Yes(36) and No(124).

Before the statistical analyses of the achievement scores were performed, a one-way analysis of variance was done with response to the Owned and Programmed a Personal Computer Before College (PCBC) question as the Independent Variable and Converted College Entrance Examination Score (CCHET) as the Dependent Variable. No significant difference was found between the mean CCHET scores of those who had Owned and Programmed a Personal Computer Before College and those who had not. On the basis of this, it was decided that there was no need to control for pretest aptitude with the CCHET. Therefore, the analysis of variance would be used as the analysis tool for this factor.

Total Test Score Analysis by PC Before College

The null hypothesis for evaluation of Total Test Achievement Score by PCBC was that there would be no significant difference between the scores of those who had Owned and Programmed a Personal Computer Before College and those had not. The Total Test Achievement Scores were analyzed by using a one-way analysis of variance with PCBC response as the Independent Variable and Total Test Score as the Dependent Variable. The result of this analysis was that there was a significant difference ($p < 0.05$) between the mean scores of the two groups. The mean score for the subjects in the PCBC-YES group was 21.61, and the mean score for the subjects in the PCBC-NO group was 19.84. Tables 38 and 39 on the next page show the detailed results of this analysis.

The difference of 1.77 between the mean scores of the two groups in favor of the PCBC-YES group indicates that this group was slightly better able to answer the test questions than the PCBC-NO group. This implies that there may be some benefit to programming a personal computer prior to taking a computing course. Because difference between the mean scores was significant, the null hypothesis must be rejected.

TABLE 40

TOTAL TEST SCORE
GROUP MEANS BY
OWNED AND PROGRAMMED A PERSONAL COMPUTER BEFORE COLLEGE

Group	Number Subjects	Mean Scores
PCBC-YES	36	21.61
PCBC-NO	124	19.84
Total	160	20.24

TABLE 41

TOTAL TEST SCORE
ANALYSIS OF VARIANCE BY
OWNED AND PROGRAMMED A PERSONAL COMPUTER BEFORE COLLEGE

SOURCE	DF	SUM OF SQUARES	MEAN OF SQUARES	F	SIG-OF-F	
Main Effect						
PCBC	1	87.65	87.65	4.24	0.04	*
Residual	158	3263.33	20.65			
TOTAL	159	3350.97	21.08			

Sub-Test Score Analysis by PC Before College

A one-way analysis of variance was performed for each of the eight sub-tests with the answer to Owned and Programmed a Personal Computer Before College as the Independent Variable and the appropriate sub-test score as the Dependent Variable. Only one of the eight sub-test analyses revealed a significant difference between the mean scores of the two groups. No significant differences were found between the mean scores of the two groups for the following sub-tests:

1. General FORTRAN Knowledge
3. Input-Output Techniques
4. Program Flow Control
5. Ability to Read Programs
6. Miscellaneous Computing Concepts
7. Use of Expressions
8. Use of Arrays

For the remaining sub-test, Knowledge of Functions and Procedures, a significant difference was found between the mean scores.

Functions and Procedures Sub-Test Analysis by PCBC

The null hypothesis tested for this sub-test was that there would be no significant difference between the mean Functions and Procedures Sub-Test Achievement

Scores of the subjects when they were grouped by whether they had owned and programmed a Personal Computer Before College. A one-way analysis of variance with PCBC response as the Independent variable and Functions and Procedures Sub-Test Score as the Dependent Variable revealed that there was a significant difference ($p < 0.05$) between the mean scores of the two groups. The mean Functions and Procedures Sub-Test score of the subjects who had owned and programmed a Personal Computer Before College was 3.75, while the mean score of the subjects who had not was 3.23. Tables 40 and 41 on the next page show the details of the results of this analysis.

The mean scores showed that the subjects who had answered PCBC-YES were able to answer 0.52 more questions correctly about functions and procedures than those who had answered PCBC-NO. Therefore, it would seem that there might be some benefit to owning and programming a personal computer prior to taking a computing course.

This analysis did not support the null hypothesis that there would be no difference between the mean scores of the two groups for this sub-test. Therefore, the null hypothesis may be rejected.

TABLE 42

FUNCTIONS AND PROCEDURES SUB-TEST SCORE
GROUP MEANS BY
OWNED AND PROGRAMMED A PERSONAL COMPUTER BEFORE COLLEGE

Group	Number Subjects	Mean Scores
PCBC-YES	36	3.75
PCBC-NO	124	3.23
Total	160	3.35

TABLE 43

FUNCTIONS AND PROCEDURES SUB-TEST SCORE
ANALYSIS OF VARIANCE BY
OWNED AND PROGRAMMED A PERSONAL COMPUTER BEFORE COLLEGE

SOURCE	DF	SUM OF SQUARES	MEAN OF SQUARES	F	SIG-OF-F	
Main Effect						
PCBC	1	7.67	7.67	4.17	0.04	*
Residual	158	290.43	1.84			
TOTAL	159	298.09	1.87			

Discussion of Personal Computer Before College Analyses

One primary and eight secondary hypotheses were tested to determine the value of owning and programming a personal computer before college as a preparation for a college computing course. They were performed for two reasons: (1) the data were easy to acquire, and (2) to discover what percentage of the participants had personal computers and to determine whether such ownership might affect performance in a college level course. The significant difference in mean Total Test scores somewhat supports such ownership. One of the eight subtopics exhibits some sensitivity to owning a personal computer with significance at the 0.05 level.

A summary of the results of testing all of these hypotheses is presented in Table 42 on the next page. In this table, the results of the analyses of variance as well as the means and standard deviations for all of the tests are provided. For six of the seven sub-tests for which no significant differences were found the actual differences are in favor of owning and programming a Personal Computer Before College. Thus, there appears to be some overall benefit which is reflected in the significant difference for the Total Test Scores.

Summary of Results of The Analysis of Variance
Of the Total Test and All Subtests With
OWNED AND PROGRAMMED A PERSONAL COMPUTER BEFORE COLLEGE
as the Independent Variable

INDEPENDENT VARIABLE VALUE	N
YES	36
NO	124

DEPENDENT VARIABLE	INDEPEND-VAR MEAN SCORE STANDARD-DEV		<----ANOVA---->		
	YES	NO	F-VAL	F-PROBAB	
Total Test Score	21.61	19.84	4.24	0.04	*
	5.07	4.38			
Genl FORTRAN Knowledge	3.06	3.01	0.06	0.81	NS
	1.04	1.04			
Functions and Procedures	3.75	3.23	4.17	0.04	*
	1.50	1.31			
Input-Output Techniques	1.81	1.75	0.07	0.80	NS
	1.19	1.13			
Program Flow Control	4.00	3.73	1.25	0.26	NS
	1.15	1.33			
Ability to Read Programs	2.61	2.59	0.01	0.92	NS
	1.23	1.22			
Misc Computing Concepts	2.67	2.37	2.66	0.10	NS
	1.04	0.93			
Use of Expressions	4.11	3.68	3.72	0.06	NS
	1.30	1.15			
Use of Arrays	1.97	2.09	0.25	0.62	NS
	1.11	1.26			

TABLE 44

CHAPTER 5

SUMMARY

CONCLUSIONS AND RECOMMENDATIONS

PURPOSE OF THE STUDY

The purpose of this study was to compare two teaching methods for teaching introductory computer programming concepts in the programming language FORTRAN-77 as measured by a retention achievement test. The comparison was to include not only the productive aspects of the methods, but also the acceptability of the experimental method as measured by the student's reactions to the methods. The experimental teaching method is a Guided-Discovery technique wherein the student is provided some guidance in the form of brief lectures on the current topic at the beginning of a two-hour laboratory session and is otherwise expected to read the text and do a comprehensive problem every week which utilizes the current material. The traditional teaching method is a Lecture-Laboratory technique in which the students are presented two one hour lectures per week and, in addition, are required to attend a two-hour laboratory session conducted by a qualified laboratory assistant in which the weekly homework assignment is discussed.

In addition to comparing the two methods, the teachers were to be compared as well. This was to provide a contrast with the direct comparison of the methods and determine to what extent the teacher might be the influencing factor as opposed to the method. The qualitative reaction of the students to the two methods was also evaluated by the study.

A secondary purpose of the study was to determine what affect certain related demographic factors might have on student success in an introductory computer programming course. Four factors were evaluated. First, the SEX of the student was used as the independent variable to determine whether there were any sexually related differences in achievement. The other three demographic factors used as independent variables were all related to the students previous experience with computers, specifically:

1. Did the student take any computing courses in high school;
2. Did the student have access to and make regular use of a computer prior to entering college; and,
3. Did the student own and program a personal computer prior to entering college?

SETTING OF THE STUDY

The Institution

The setting of the study was the Colorado School of Mines in Golden, Colorado. The Colorado School of Mines is a four-year engineering school which grants baccalaureate, masters and doctoral degrees in mineral and energy industry related scientific and engineering disciplines. The student body consists of approximately 2000 undergraduates students, and 500 graduate students. This school is noted for having the highest entrance requirements of any publically operated school in the country.

The Population

The segment of the Colorado School of Mines student population which participated in the study was the Freshman Class that entered the school in the Fall of 1984 and enrolled in either EP-101 Engineering Practices, Introductory Course Sequence (EPICS), or MA-115 Introduction to Computer Programming. The Specific subjects of the study were those Fall 1984 Freshmen who returned as Sophomores in the Fall of 1985, and enrolled in either Calculus II or Calculus III, and who had a recorded college entrance examination score.

The EPICS Program

EPICS was an experimental program in which the three disciplines of Engineering, English and Computer Science developed an integrated course designed to prepare the student to succeed in school as well as in the workplace. Assignments were developed and evaluated jointly by the three departments. Each EPICS section met once each week for two hours with each of the three discipline instructors.

REVIEW OF METHODS AND PROCEDURES

Selection of Subjects

The subjects were divided into two groups by randomly selecting 110 American students and 10 foreign students from a pool of volunteers for the Guided-Discovery experimental group, and assigning the balance of the students, approximately 270 subjects, into the traditional Lecture-Laboratory control group. Both the experimental group and the control group were divided into three sections which met a different times. The same teacher taught all three sections of the experimental group. Each of the three sections of the control group was taught by a different teacher laboratory instructor team. Both groups were presented the same material from the same book. However, there were differences in the order of presentation between the two groups, but not within the sections of the groups.

The Method of the Study

The method of conducting the study was to perform a post hoc analysis of the two teaching methods by measuring student retention of introductory computer programming concepts with an achievement test. The achievement test was derived from the textbook and the lesson plans for the two courses. Pre-treatment aptitude was controlled by using a converted college entrance examination score as a covariate.

The Treatment

The main difference between the two groups was the treatment they received. The Guided-Discovery group was presented with a brief introductory lecture at the beginning of a two hour laboratory session in which the subject of the week was introduced and related to the activities in the other two disciplines participating in the EPICS program. The students were expected to have read the assigned material and to be prepared to apply it to the current problem during the meeting period. The Lecture-Laboratory groups each met twice per week for a typical one-hour lecture and once each week for a two-hour laboratory session in which they applied what they had been taught.

The Test Instrument

The test instrument was a 40-question multiple choice test on the material covered. It consisted primarily of questions on the FORTRAN-77 Programming language. There were a few questions on material presented from the text which was of a general nature, but which had been explained in terms of the programming language. The test was developed with the assistance of all of the instructors of the two groups.

In addition to the substantive questions, several demographic questions relating to the student's previous experience with computers were included in the test instrument. This was done to determine what affect, if any, these factors might have on success in an introductory computer programming course.

Collecting the Data

The data were collected by testing all of the students in Calculus II and Calculus III on the same day at the beginning of the Fall 1986 semester. The test instrument was administered by the Calculus teachers. The date for the test was chosen to be early in the term to prevent contamination of the results by the subjects participation in new computing courses.

CONCLUSIONS

For a setting consisting of an engineering oriented school with a highly motivated student population, and based upon the findings of the study, the following conclusions have been reached:

1. General achievement in an introductory computer programming course is not affected by the teaching method used when the choice of method is between the Guided-Discovery method and a more traditional Lecture-Laboratory method
2. It is expected that students who take introductory computer programming under the Guided-Discovery method are more likely to achieve higher scores in certain topical areas than students who are taught by more conventional methods. Three specific topical areas in which Guided-Discovery students may gain in achievement over their Lecture-Laboratory counterparts are:
    a. Input-Output Techniques;
    b. Ability to Read Programs; and,
    c. Use of Arrays

    However, when differentiating between the teacher who taught the section and the teaching method used, only The Ability to Read Programs appears to be uniquely affected by the Guided-Discovery method. This would seem to be a logical consequence of requiring the student to be essentially self-taught which would require the reading of a large number of programs.

3. The Guided-Discovery method is perceived to be fully equivalent to more traditional methods by the students and is equally acceptable to them.
4. General achievement in an introductory computer programming course is not affected by the teacher who teaches the course.
5. The sex of the student is not related to achievement in an introductory computer programming course.
6. Factors which may affect achievement in an introductory computer programming course are
    a. Taking a High School Computing Course;
    b. Making Regular use of a Computer Prior to Taking the Course; and,
    c. Owning and Programming a Personal Computer Prior to Taking the Course.

In general, any prior experience with a computer is a better preparation than none, and the more formal the experience, the greater the gain.

RECOMMENDATIONS

Recommendations for Colorado School of Mines

Adoption of the Guided-Discovery method and the EPICS program is a viable and desirable course of action. The students perceive it to be equivalent to more traditional methods and, therefore acceptable. It will probably be necessary to make some provision for a detailed explanation of the content of the EPICS program for students who elect to transfer to another institution after their Freshman year.

Recommendations for Computer Programming Students

Most colleges and universities now require all students to take some introductory computing course if only to acquaint them with potential uses of computers. This study has found that prior exposure to computers either in a formal class or in informal use has been shown to be of positive benefit to achievement in a college computing course. Therefore, high school students who are planning to enter college would be well advised to take a computing course in high school regardless of what they plan to study in college. Even students who are not planning to go to college could benefit in the workplace by taking such a course.

Recommendations for Further Research

The conduct of this study has raised several questions which may be answered by further research. The unique inter-disciplinary nature of the EPICS program at the Colorado School of Mines suggests several topics of interest. Likewise, the results attributable to prior exposure to computers lead to interesting questions. The following suggestions are only those which would be of the most immediate benefit.

1. Studies should be conducted in which the complete gain to the student in a multi-disciplinary course is measured rather than in one subject area.
2. Other high-technology subjects should be tested to determine whether they are also amenable to being taught in an inter-disciplinary environment regardless of the teaching method used.
3. Most research related to teaching methods centers around a search for a method which produces greater educational benefit for the student. There is a need to investigate innovative teaching methods which are more economical for the institution to use and which are acceptable to the students.

4. Since this experiment was conducted in a highly restricted environment which limits the extent to which the results can be generalized, it should be replicated in greatly different environments with different student populations. For example, if similar results were found for a liberal arts school or for a regional state college in a rural setting, or for students who are not typically self-motivated the results would be more general.
5. One finding of this study was that Owning and Programming a Personal Computer Before Entering College had a positive influence on achievement in an introductory computer programming course. However, this factor was not as influential as Taking a Computing Course in High School. It would seem that an investigation of the difference between these two factors might lead to some interesting results.
6. The study made by Dubin and Taveggia is many years old and was extremely general and therefore subject to criticism. A similar study which would collect and catalog research on teaching methods, or experiments whose motivation was economics would be useful to administrators.

IMPLICATIONS OF THE STUDY

It would seem that this study has shown, yet once again, that the student is most often the controlling factor in the educational process. This is perhaps best said by the famous western writer Louis L'Amour in the epic novel "The Lonesome Gods" in the following quotation from a conversation between a teacher and his pupils, one of whom was the protagonist:

> "Actually," he said one morning, "all education is self-education. A teacher is only a guide, to point out the way, and no school, no matter how excellent, can give you an education.
>
> "What you receive is like the outlines in a child's coloring book. You must fill in the colors yourself." [L'Amour, 1983]

This is as good an explanation of the guided-discovery method as any other. Perhaps the Colorado School of Mines EPICS program carries the concept beyond what is intended by Mr. L'Amour, but the students of the Colorado School of Mines are beyond what might be termed typical college students by any measure.

The problem to be resolved was finding a more cost effective way to teach introductory high-technology courses to entering college students. A related problem was that the more cost effective method should be accepted by the students as equivalent to or better than the normal alternative. A solution, for the

Colorado School of Mines, was to implement an innovative multi-disciplinary guided-discovery course in which the emphasis was on student self-instruction. The questions were, "Will the students do as well as they have in more traditional classes," and "Will the students accept the new technique." The answers to both questions based on the findings of this study are "Yes!"

The implications of this study are that innovative and more cost effective teaching techniques can be used to advantage in the college classroom without adversely affecting the quality of the educational product. Whether the motive is to reduce the direct costs to the institution or to make more effective use of limited resources or to produce better educated students, as long as the students perceive an equivalent benefit they will accept the innovative method as equivalent to that with which they are more familiar.

Teachers of computer science and other high cost or high technology subjects should experiment to find better more productive and cost effective ways to teach their classes.

REFERENCES

Anderson-Bell (1984). *ABSTAT, Release 4, User's Guide*. Canon City, Colorado: Anderson-Bell.

Appleby, Doris C. (1986, June). And Still More on Retraining Mathematics Faculty to teach Undergraduate Computer Science. *The Association for Computing Machinery, Special Interest Group for Computer Science Education Bulletin, 18*(2), 17-24.

Ayen, Wm. E., and Grier, S. (1983, February). A New Environment for Teaching Computer Science. In Sheau-Dong Lang (Ed.), *The Papers of the Fourteenth SIGCSE Technical Symposium on Computer Science Education, The Association for Computing Machinery, Special Interest Group on Computer Science Education Bulletin 15*(1), 258-264.

Ballew, David (1985, December). More Thoughts on the Need for Retraining to Teach Undergraduate Computer Science. *The Association for Computing Machinery, Special Interest Group on Computer Science Education Bulletin 17*(4) 17-24.

Calamari, Mary Diane Bodin, Ph.D. (1983). A Comparison of Two Methods of Teaching Computer Programming to Secondary Mathematics Students. *Dissertation Abstracts International, 45*, 67-A. (University Microfilms No. 84-09, 574)

Chew, Robert L., Ph.D. (1985, February). Bucking the Tide: A Transition form Industry to Academe. In Lillian N. Cassel and Joyce Currie Little (Eds.), *The Papers of the Fifteenth SIGCSE Technical Symposium on Computer Science Education, The Association for Computing Machinery, Special Interest Group on Computer Science Education Bulletin 16*(1) 5-9.

Clark, Thomas Blair, Ph.D. (1975). A Comparison of Two Approaches to Teaching Computer Programming to Prospective Elementary School Teachers. *Dissertation Abstracts International, 36*(6), 3577-A. (University Microfilms No. 75-26, 761)

Davis, James R. (1976). Teaching Strategies for the College Classroom. Boulder Colorado: Westview Press.

Drew, Mark S. and Caplin, Shane D. (1984, June). Batch LOGO -- A Strategy for Introducing PL/I and Structured Programming to Gifted High School Students. The Association for Computing Machinery, Special Interest Group on Computer Science Education Bulletin 16(2), 13-16.

Dubin, Robert, and Taveggia, Thomas C. (1968). The Teaching-Learning Paradox: A Comparative Analysis of College Teaching Methods. Eugene Oregon: University of Oregon Press

Duggar, Cynthia Miller, Ph.D. (1983). A Study of the Relationships Among Computer Programming Ability, Computer Program Content, Computer Programming Style, and Mathematical Achievement in a College Level BASIC Programming Course. Dissertation Abstracts International, 44(1), 95-A. (University Microfilms No. 83-11, 672)

Dyck, V. A., Lawson, J. D., Smith, J. A. (1984). FORTRAN/77: An Introduction to Structured Problem Solving. Reston, Virginia: Reston Publishing Company, Inc.

Evangelauf, J. (1984, May 14). Professors in High-Demand Fields Are Getting Higher-Than-Average Salaries, 2 Studies Find. The Chronicle of Higher Education, pp 25-28.

Ferguson, George A. (1981). Statistical Analysis in Psychology and Education (Fifth Edition). New York: McGraw-Hill.

Gathers, Emery (1986, September). Screening Freshmen Computer Science Majors. The Association for Computing Machinery, Special Interest Group on Computer Science Education Bulletin 18(3) 44-48.

Grossman, Elaine Sue, Ed.D. (1983/1984). A Comparison of Instructional Methods Employing Group and Individual Programming in an Introductory Computer Science Course. Dissertation Abstracts International, 44(8), 2397-A. (University Microfilms No. 83-22, 206).

Hall, Dale Junior, Ed.D. (1965). An Exploratory Study of Self-Instruction Text and Lecture Systems of Teaching FORTRAN Computer Programming. Dissertation Abstracts International, 26(9), 5231-5232. (University Microfilms No. 65-14, 040)

Haring, Ed (1985). Teaching and Learning Styles (Technical Report). Elgin, Illinois: Elgin Community College. (EDRS No. ED-258-658)

Heeler, Phillip J. (1983, February). A Master's Degree in School Computer Studies. In Sheau-Dong Lang (Ed.), The Papers of the Fourteenth SIGCSE Technical Symposium on Computer Science Education, The Association for Computing Machinery, Special Interest Group on Computer Science Education Bulletin 15(1) 99-103.

L'Amour, Louis (1984). The Lonesome Gods (Bantam Rack-size Edition, p 165). New York: Bantam Books.

Kachigan, Sam Kash (1986). Statistical Analysis: An Interdisciplinary Introduction to Univariate & Multivariate Methods. New York: Radius Press.

Karian, Zaven (1984, February). A Central Ohio Consortium for Retraining in Computer Science. In Lillian N. Cassel and Joyce Currie Little (Eds.), The Papers of the Fifteenth SIGCSE Technical Symposium on Computer Science Education, The Association for Computing Machinery, Special Interest Group on Computer Science Education Bulletin 16(1), 2-4.

Martin, J. B., and Martin, K. E. (1986, February). A Profile of Today's Computer Literacy Student. In Lillian N. Cassel and Joyce Currie Little (Eds.), The Papers of the Seventeenth SIGCSE Technical Symposium on Computer Science Education, The Association for Computing Machinery, Special Interest Group on Computer Science Education Bulletin 18(1), 27-33.

Martin, Kenneth E. (1984, September). Student Performances and Student Perceptions of 'Principles of Computer Programming'. The Association for Computing Machinery, Special Interest Group on Computer Science Education Bulletin 16(3), 29-33.

McEntyre, Diane Sorben, Ph.D. (1977/1978). Structures and Strategies for Teaching Computer Programming in an Introductory Course. Dissertation Abstracts International, 38(08), 4713-A. (University Microfilms No. 77-31, 452).

McLaughlin, Brian, Ph.D. (1981). An Experimental Comparison of Discovery and Didactic Computerized Instructional Strategies in the Learning of Computer Programming. Dissertation Abstracts International, 42(2), 529-A. (University Microfilms No. 81-16, 843)

Miller, Nancy Ellen, Ph.D. (1981/1982). The Effect of Team Programming on Student Achievement in COBOL Instruction. Dissertation Abstracts International 42(7), 3000-A. (University Microfilms No. 81-28, 840).

Mitchell, William, Ph.D. (1983, February). Retraining: Is It The Answer to the Computer Faculty Shortage. In Sheau-Dong Lang (Ed.), The Papers of the Fourteenth SIGCSE Technical Symposium on Computer Science Education, The Association for Computing Machinery, Special Interest Group on Computer Science Education Bulletin 15(1) 89-98.

Mitchell, William, Ph.D., and Hartman, Janet (1985, March). Summer Faculty Retraining in Computing, A report on Four Programs. In Harriet G. Taylor (Ed.), *The Papers of the Sixteenth SIGCSE Technical Symposium on Computer Science Education, The Association for Computing Machinery, Special Interest Group on Computer Science Education Bulletin 17*(1), 289-291.

Mitchell, William, Ph.D. (1986, February). Retraining Computing Faculty, A Perspective. In Lillian N. Cassel and Joyce Currie Little (Eds.), *The Papers of the Seventeenth SIGCSE Technical Symposium on Computer Science Education, The Association for Computing Machinery, Special Interest Group on Computer Science Education Bulletin 18*(1), 203-210.

Narthasilpa, Amnuay, Ph.D. (1984). The Effects of Microcomputer Instruction on Knowledge in Computer Programming and Attitudes of Science Education Students. *Dissertation Abstracts International, 45*(9), 2826-A. (University Microfilms No. 84-19, 644)

Norusis, Marija J. (1985). *Advanced Statistics Guide to SPSSx*. Chicago: McGraw-Hill.

Payne, Harry E., Jr. (1983). The Effects of Three Instructional Techniques on the Problem-Solving Ability of General Education Mathematics Students at the Junior College Level (Research Report). Florida College. (EDRS No. ED-262-979, 1983).

Plog, Claudia Elizabeth (1980/1981). The Relationship of Selected Variables in Predicting Academic Success in Computer Programming. *Dissertation Abstracts International, 41*(7), 2903-A. (University Microfilms No. 80-27, 678).

Pommersheim, John Paul, Ph.D. (1983). Relationships Between Individual Cognitive Profiles and Achievement in Computer Programming. *Dissertation Abstracts International, 45*, 444-A. (University Microfilms No. 84-11, 827)

Ramberg, Peter (1986, September). A New Look at an Old Problem: Keys to Success for Computer Science Students. *The Association for Computing Machinery, Special Interest Group on Computer Science Education Bulletin, 18*(3), 36-37.

Reisman, Sorel, Ph.D. (1973/1978). The Effect of Selected Variables on the Learning of Some Computer Programming Elements. *Dissertation Abstracts International, 38*(06), 3257-A.

Ricardo, Catherine Mary, Ph.D. (1983). Identifying Student Entering Characteristics Desirable for a First Course in Computer Programming. *Dissertation Abstracts International, 44*(1), 96-A. (University Microfilms No. 83-11, 856).

Rusnock, Margaret I., Ph.D. (1983/1984). The Effect of Student Cognitive Profile on Success in Computer Programming and Attitude Toward Computer Programming. *Dissertation Abstracts International, 45*(02). 44-A. (University Microfilms No. 84-11, 693).

Ryder, B. G. (1984, February). A 'Hands-On' Approach to Computer Literacy. In Lillian N. Cassel and Joyce Currie Little (Eds.), *The Papers of the Fifteenth SIGCSE Technical Symposium on Computer Science Education, The Association for Computing Machinery, Special Interest Group on Computer Science Education Bulletin 16*(1), 102-107.

Scanlan, David (1985, March). Retraining: Master's Degree Proposal for Applied Computer Science Education for Community College Teachers. In Harriet G. Taylor (Ed.), *The Papers of the Sixteenth SIGCSE Technical Symposium on Computer Science Education, The Association for Computing Machinery, Special Interest Group on Computer Science Education Bulletin 17*(1), 289-291.

Schmalz, Rosemary (1985, September). Some Thoughts on Retraining and the Lack Thereof of a Mathematics Educator Teaching Computer Science. *The Association for Computing Machinery, Special Interest Group on Computer Science Education Bulletin, 17*(3), 17-23.

Sharma, Onkar P., Ph.D. and Behforooz, Ali, Ph.D. (1985, March). Retraining for a Graduate Program in Computer Science. In Harriet G. Taylor (Ed.), The Papers of the Sixteenth SIGCSE Technical Symposium on Computer Science Education, The Association for Computing Machinery, Special Interest Group on Computer Science Education Bulletin 17(1), 282-288.

Snedecor, George W, and Cochran, William G. (1979). Statistical Methods, (Sixth Edition). Ames, Iowa: Iowa State University Press.

SPSS Inc. (1983). Users Guide to SPSSx. Chicago: McGraw-Hill

Taulbee, O. E. (1986, September). Annual U.S. Summaries of Ph.D. Production and Employment in Computer Science 1970-1985. Association for Computing Machinery, Special Interest Group on Computer Science Education Bulletin, 18(3), 2-8.

Taulbee, O. E. and Conte, S. D. (1979, February). Production and Employment Of Ph.D.s in Computer Science -- 1977-78. Communications Of The Association for Computing Machinery, 22(2), 75-76.

Taulbee, O. E. and Conte, S. D. (1977, June). Production and Employment Of Ph.D.s in Computer Science -- 1976. Communications Of The Association for Computing Machinery, 20(6), 370-372.

Taulbee, O. E. and Conte, S. D. (1976, June). Production and Employment Of Ph.D.s in Computer Science. Communications Of The Association for Computing Machinery, 19(6), 311-313.

Thronson, Roderick Morris, Ed.D. (1984/1985). Achievement as a Function of Learning Style Preference in Beginning Computer Programming Courses. Dissertation Abstracts International, 45(10), 3100-A. (University Microfilms No. 85-00, 623).

Werth, Laurie Honour (1986, February). Predicting Student Performance in a Beginning Computer Science Class. In Lillian N. Cassel and Joyce Currie Little (Eds.), *The Papers of the Seventeenth SIGCSE Technical Symposium on Computer Science Education, The Association for Computing Machinery, Special Interest Group on Computer Science Education Bulletin 18*(1), 138-142.

Whipkey, Kenneth L. (1984, December). Identifying Predictors of Programming Skill. *The Association for Computing Machinery, Special Interest Group on Computer Science Education Bulletin, 16*(4), 36-42.

Wiggins, Timothy A., Ph.D. (1984). The Effect of Teaching Method or Student Characteristics on Student Achievement or Attitude in a BASIC Computer Programming Undergraduate Course in Agricultural Mechanization at Iowa State University. *Dissertation Abstracts International, 45*(7), 1962-A. (University Microfilms No. 84-23, 752)

Witherell, Robert Joseph, Ed.D. (1979). An Analysis of the Effectiveness of Methods of Instruction Used in the Teaching of FORTRAN IV Computer Programming. *Dissertation Abstracts International, 40*(9), 4860-A. (University Microfilms No. 80-05, 272)

APPENDIX A

THE TEST INSTRUMENT

TO: The

FROM: Ch

The
purposes

1.
pa
st
ho

2.
res
Phi

You
beginnin
only to
Registra
or SAT
soon a
matricul
anyone f
from co
seperatel

The
as the q
will hel
to other
identify
some spec
method
grades
computer

The
results,
student
to my o
results.
shows you

Plea
ability.
you know
give it
circling
on the
since all

Than

TO: The students participating in this experiment

FROM: Charles P. Howerton, Assoc. Prof. Computer Science

The experiment in which you are about to participate has two purposes:

1. By comparing the results from the students who participated in the EPICS program with the results from the students who did not participate in the EPICS program, it is hoped that the EPICS program can be improved where needed.

2. The preparation of this test, and the analysis of the results fulfill part of the requirements for my Doctor of Philosophy Degree.

You are asked to record your matriculation number at the beginning of the personal information section. This will be used only to cross reference to the files in the Office of the Registrar. The cross reference will be used to record your ACT or SAT scores which will be used in analysing the results. As soon as the ACT and SAT scores have been recorded your matriculation number will be replaced by a code number to prevent anyone from knowing how well you did. A cross reference listing from code number to matriculation number will be maintained seperately, by me personally, and will be shown to no one.

The answers you give to the other personal questions, such as the questions about your previous experience with computers, will help in the analysis of the results. The answers you give to other questions, such as who was your instructor, will help to identify those instructors who were particularly good at teaching some specific concept or topic so that they can help develop the method for teaching it in the future. The questions about your grades will help to develop a correlation between calculus and computer programming.

The <u>only</u> statistics which will be made public will be group results, such as xx% of the students missed question 34. Any student who wishes to know how he or she did personally may come to my office after the Christmas holiday and I will report the results. To get the results you must have your student ID which shows your matriculation number.

Please answer all of the questions to the best of your ability. Please do not make any <u>wild</u> quesses. If you think that you know the answer but are not absolutely sure then by all means give it your best shot. If you do not know, then say so by circling answer "e" (I do not know). There are a few questions on the test which not everyone will be able to answer correctly since all classes were not taught precisely the same material.

Thank you for your participation in this experiment.

A. Enter matriculation number ____________________

B. Enter date of birth ____/____/____
MONTH DAY YEAR

C. SEX (circle one)
   a. FEMALE
   b. MALE

D. Did you take any computing courses in High School?
   a. Yes
   b. No

E. Did you have regular access to and make use of a computer prior to entering Colorado School of Mines?
   a. Yes
   b. No

F. Do you or your family own a personal or home computer for which you wrote programs prior to taking your first computing course at CSM?
   a. Yes
   b. No

G. Which of the following was your first college level computing course?
   a. EPICS ------------> GOTO question H
   b. MA-115 ------------> GOTO question M
   c. EG-101 ------------> GOTO question Q
   d. other at CSM, course number __________ --> Go to U.
   e. Not at CSM, course title ______________--> Go to U.

EPICS STUDENTS ONLY - QUESTIONS H THRU L

EPICS STUDENTS ONLY
H. Who was your EPICS FORTRAN Computing Instructor?
   a. Jean Bell
   b. Other, enter name ____________________________

EPICS STUDENTS ONLY
I. If you took EPICS in the Spring semester of 1985 on what day of the week did you have the FORTRAN computer programming portion of your EPICS class?
   a. Monday
   b. Wednesday
   c. Friday
   d. I did not take EPICS in the Spring of 1985

EPICS STUDENTS ONLY
J. If you took EPICS FORTRAN Programming in the Spring semester of 1985 did your group use the DEC-10 first or the TIPC first?
   a. I used the DEC-10 before I used the TIPC
   b. I used the TIPC before I used the DEC-10
   c. I did not take EPICS in the Spring of 1985

EPICS STUDENTS ONLY

K. What grade did you receive in the Spring semester of 1985 in EPICS?

a. A	d. D
b. B	e. F
c. C	

EPICS STUDENTS ONLY

L. How do you feel that the FORTRAN computer programming portion of the EPICS program compared to the more traditional approach used in MA-115 or EG-101?
- a. Much Better
- b. Better
- c. About the same
- d. Worse
- e. Much Worse
- f. I do not know anything about MA-115 or EG-101

EPICS STUDENTS GO TO QUESTION U

MA-115 STUDENTS ONLY - QUESTIONS M THRU P

MA-115 STUDENTS ONLY

M. Who was your MA-115 instructor?
- a. Charles P. Howerton
- b. Patrick Madison
- c. Donald Marsh
- d. Craig Murphy
- e. Other, enter name ______________________________

MA-115 STUDENTS ONLY

N. When did you take MA-115?
- a. Fall 1984
- b. Spring 1985
- c. Summer 1985

MA-115 STUDENTS ONLY

O. What grade did you receive in MA-115?

a. A	d. D
b. B	e. F
c. C	

MA-115 STUDENTS ONLY

P. How do you feel that the computer programming portion of the EPICS program compared to the more traditional approach used in MA-115?
- a. Much Better
- b. Better
- c. About the same
- d. Worse
- e. Much Worse
- f. I do not know anything about the EPICS program

MA-115 STUDENTS GO TO QUESTION U

EG-101 STUDENTS ONLY - QUESTIONS Q THRU T

EG-101 STUDENTS ONLY
Q. Who was your EG-101 instructor?
a. Enter Name ______________________________

EG-101 STUDENTS ONLY
R. When did you take EG-101?
a. 1982-1983 Academic Year
b. 1983-1984 Academic Year
c. 1984-1985 Academic Year
d. Other, specify ______________________________

EG-101 STUDENTS ONLY
S. What grade did you receive in EG-101?
a. A d. D
b. B e. F
c. C

EG-101 STUDENTS ONLY
T. How do you feel that the computer programming portion of the EPICS program compared to the more traditional approach used in EG-101?
a. Much Better
b. Better
c. About the same
d. Worse
e. Much Worse
f. I do not know anything about the EPICS program.

****

ALL STUDENTS ANSWER QUESTIONS U THRU Z

ALL STUDENTS ANSWER
U. Have you taken any computer programming courses since you took EPICS, or MA-115 or EG-101, or the introductory course?
a. YES
b. NO

ALL STUDENTS ANSWER
V. What was your grade in your <u>first</u> college calculus course?
a. A d. D
b. B e. F
c. C

For all of the questions in this test, you may assume that the following declarations have been made, if any other variable identifiers are used assume that normal default typing is in effect.

```
        REAL    X, Y, Z
        INTEGER I, J, K
        LOGICAL P, Q
```

<<<***** CIRCLE THE LETTER CORRESPONDING TO YOUR ANSWER *****>>>

1. A STOP statement
   a. Terminates the execution of a FORTRAN program
   b. Is the last statement in a FORTRAN program
   c. May not be used in an IF statement
   d. Indicates the end of a DO loop
   e. I Do not know

2. An array name without a subscript may appear in a program:
   a. In an argument list of a SUBROUTINE
   b. In an arithmetic expression
   c. In a logical expression
   d. If it is the control variable of a DO loop
   e. I Do not know

3. The LABEL which follows the word DO at the beginning of a DO loop indicates:
   a. Where to go to when the DO loop is finished
   b. The last statement in the range of the DO loop
   c. Where to go if an error occurs in the DO loop
   d. Has no specified purpose
   e. I Do not know

4. Which of the following is a valid arithmetic assignment statement?
   a. DATA X, Y /2*0.0/
   b. X = X + 1
   c. ASSIGN 25 TO JOE
   d. READ(5, 20) X
   e. I Do not know

5. Given the following statement, how many elements are in the array M?
   REAL M(10)
   a. Depends on how M is used
   b. 11
   c. 10
   d. M cannot be a REAL array name
   e. I Do not know

6. Arrays are dimensioned in a:
   a. DIMENSION statement
   b. Type statement
   c. Both a and b
   d. Neither a nor b
   e. I Do not know

7. Given the following program fragment, what value will be stored in Z?

```
X = 2.0
Y = 6.0
Z = 3.*Y + 2./X
```

a. 10
b. 12
c. 19
d. None of the Above
e. I Do not know

8. The control variable of a DO statement:
a. Is always a subscripted variable
b. Is assigned a value by the DO statement
c. Is incremented using an arithmetic assignment statement
d. Is a LOGICAL variable
e. I Do not know

9. The END statement:
a. Is the last executable statement in a FORTRAN program
b. Marks the end of a DO statement range
c. Terminates every FORTRAN program unit
d. Is only used at the end of the main program
e. I Do not know

10. An argument of a FUNCTION may be:
a. A constant
b. A variable
c. An expression
d. All of the above
e. I Do not know

11. A logical expression may contain:
a. Arithmetic operators
b. REAL constants
c. Logical operators
d. All of the above
e. I Do not know

12. Which of the following is an unconditional transfer statement?
a. IF (A .LE. B) GOTO 100
b. IF (A .NE. B) STOP
c. GOTO 100
d. DO 100
e. I Do not know

13. When you compile a program, you:
a. Translate FORTRAN statements into machine language
b. Draw a flowchart of the program
c. Write the FORTRAN statements using a flowchart or pseudocode as a guide
d. Debug a program from a listing of the program
e. I Do not know

14. Which of the following FORMAT statements could be used to output a line of a table consisting of an integer followed by two real numbers?
a. FORMAT ´X, 3I5´
b. FORMAT (X,3I5)
c. FORMAT ´X,I5,2F10.6´
d. FORMAT (X,I5,2F10.6)
e. I do not know

15. Which of the following will always return exactly one value?
a. FUNCTION subprogram
b. RETURN statement
c. SUBROUTINE subprogram
d. DATA statement
e. I Do not know

16. Which of the following can not be the identifier or name of an INTEGER variable?
a. J
b. K
c. J.3
d. INCOME
e. I Do not know

17. Which of the following arithmetic statements is incorrect?
a. I = A + B
b. X(J + 2) = X(2)
c. Z = 2*I + J
d. A = X / -17.5
e. I Do not know

18. In the following input statement the 5 refers to:
READ (5,7) A, B, C, I, J
a. The number of values to be input
b. A FORMAT statement label
c. An input device logical unit number
d. The number of records to be input
e. I Do not know

19. Which of the following is a correctly written logical IF?
a. IF (A - B) 10, 10, 20
b. IF (A .GT. B) STOP
c. IF (A .EL. B) GO TO 100
d. All of the above
e. I Do not know

20. Which of the following parts of a loop is not a component of the DO statement?
a. Initializing the control variable
b. Performing the process
c. Incrementing the control variable
d. Testing the control variable against the limits
e. I Do not know

21. Which of the following can be used as the terminal statement in a DO loop?
    a. A DO statement
    b. A CONTINUE statement
    c. An assignment statement
    d. b and c
    e. I Do not know

22. What is the function of the RETURN statement?
    a. To send a value to another routine
    b. To exit a DO loop before the limit is reached
    c. To end the source code of a subroutine or function
    d. To perform an orderly exit from a subroutine or function
    e. I Do not know

23. Given the following pseudo code WHILE loop

```
      WHILE ( I < 5 ) DO
             perform computations
```

Which program fragment is the best simulation?

```
a.    100  IF ( I < 5 ) THEN
                  perform computations
                  GO TO 100
           ENDIF

b.    100  CONTINUE
                  perform computations
           IF ( .NOT. I < 5 ) GO TO 100

c.    100  IF ( I < 5 ) GOTO 200
                  perform computations
           GOTO 100
      200  CONTINUE

d.         DO 100 I=1,5
                  perform computations
      100  CONTINUE
```

e. I Do not know

24. In the following statement, the number 2 is:

```
      DO 5 J = 1, 10, 2
```

a. The initial value
b. The terminal statement label
c. The increment
d. The test value
e. I Do not know

25. Which of the following is a relational operator?
    a. .LE.
    b. .NOT.
    c. *
    d. .TRUE.
    e. I Do not know

26. An implied DO:
a. Is always used with subscripted variables
b. Can be used to READ or WRITE arrays
c. Is found in a FORMAT statement
d. Is the process part of a WHILE (condition) DO loop
e. I Do not know

27. Which of the following cannot be a subscript?
a. I + J
b. P
c. 3*J + 5
d. K - 7943
e. I Do not know

28. Given the following program fragment, what will be the value of P?

```
      I = 5 / 2
      P = ( I .GT. 2 )
```

a. .TRUE.
b. .FALSE.
c. 2.5
d. 'TRUE'
e. I Do not know

29. Given the following SUBROUTINE subprogram:

```
      SUBROUTINE MYSTRY(A,B,C)
      REAL A,B,C
      C = A + B
      IF ( A .GT. B ) THEN
            C = A
            A = B
            B = C
      ENDIF
      RETURN
      END
```

and the following program fragment:

```
      X = 8.8
      Y = 7.6
      CALL MYSTRY(X,Y,Z)
```

What will be the value of Z after returning from the subroutine?
a. 8.8
b. 7.6
c. 16.4
d. 1.2
e. I Do not know

30. Which of the following will give the remainder from the integer division of I by J?
a. I - I / J
b. REMDR(I/J)
c. MOD(I,J)
d. REMAIN(I,J)
e. I Do not know

31. The following pseudocode FOR statement is best coded in FORTRAN-77 by:

```
for I from 1 to 10 by 2
      stuff to do
```

a.

```
      I = 1
100   IF ( I .NE. 10 ) THEN
            stuff to do
            I = I + 2
            GOTO 100
      ENDIF
```

b.

```
      I = 1
100   IF ( I .LT. 12 ) THEN
            stuff to do
            I = I + 2
            GOTO 100
      ENDIF
```

c.

```
      DO 100 I=1,10,2
            stuff to do
100   CONTINUE
```

d.

```
      FOR I = 1 TO 10 STEP 2
            stuff to do
      NEXT I
```

e. I Do not know

32. Given the following, what is the function of the number 100?

```
READ (5,*,END=100) X
```

a. If X is an array, X(100) is the last element to read
b. 100 is the maximum number of times this read can be done
c. 100 is the label of the statement to go to at end of file
d. 100 is the label of the CONTINUE statement at the end of the DO loop controlling the READ
e. I Do not know

33. Which of the following data types can be returned by a FUNCTION subprogram?
a. INTEGER and REAL
b. CHARACTER
c. LOGICAL
d. All of the above
e. I Do not know

34. What data type would the following expression produce?

```
(4.0 * 3.14159 * 2.5 ** 2.0) / 3.0
```

a. CHARACTER
b. INTEGER
c. LOGICAL
d. REAL
e. I do not know

35. Given the following type definition:

```
     INTEGER MYARAY(12,12)
```

and keeping in mind that FORTRAN stores arrays in column major order, which of the following SUBROUTINES should be used to read data into a row of the array?

a.

```
          SUBROUTINE RDAROW(I)
          INTEGER I,J
          DO 100 J=1,12
               READ (*,*) MYARAY(J,I)
      100 CONTINUE
          END
```

b.

```
          SUBROUTINE RDAROW(I,A)
          INTEGER I,J,A(12,12)
          DO 100 J=1,12
               READ (*,*) A(J,I)
      100 CONTINUE
          END
```

c.

```
          SUBROUTINE RDAROW(I)
          INTEGER I,J
          DO 100 J=1,12
               READ (*,*) MYARAY(I,J)
      100 CONTINUE
          END
```

d.

```
          SUBROUTINE RDAROW(I,A)
          INTEGER I,J,A(12,12)
          DO 100 J=1,12
               READ (*,*) A(I,J)
      100 CONTINUE
          END
```

e. I Do not know

36. Which of the following will always work when testing two REAL numbers for equality to the nearest thousandth?

a. `IF ( X .EQ. Y )`
b. `IF ( (X - Y) .GT. 0.001 )`
c. `IF ( ABS(X-Y) .LT. 0.001 )`
d. `TEST(X,Y,0.001)`
e. I Do not know

37. A bubble sort:

a. Is easy to code, and acceptable for short lists
b. Is used to sort champagne bottles for shipping
c. Uses a stack to simulate recursion
d. Is faster than QUICKSORT
e. I Do not know

38. The following SUBROUTINE will:

```
          SUBROUTINE DOIT(X,IX,JX)
          REAL X(IX,JX)
          INTEGER IX,JX,I,J
          DO 100 I=1,IX
               WRITE (*,*) (X(J,I),J=1,JX)
      100 CONTINUE
          END
```

a. Write out the array X one column per line
b. Write out the array X one row per line
c. Write out the array X all on one line
d. Write out the main diagonal of the array X
e. I Do not know

39. Given the following declaration:
      REAL A(20)
Which of the following program fragments will always set X to the smallest value in the one-dimensional REAL array A?

a.

```
          X = 0.0
          DO 100 I = 1, 20
               IF ( A(I) .LT. X ) THEN
                    X = A(I)
               ENDIF
      100 CONTINUE
```

b.

```
          X = 1.0E38
          DO 100 I = 1, 20
               IF ( X .LT. A(I) ) THEN
                    X = A(I)
               ENDIF
      100 CONTINUE
```

c.

```
          X = SMALLEST(A,20)
```

d.

```
          X = A(1)
          DO 100 I = 2, 20
               IF ( A(I) .LT. X ) X = A(I)
      100 CONTINUE
```

e. I Do not know

40. A binary search is so named because it:
a. Is used when looking for binary numbers
b. Searches by comparing alternating pairs of numbers
c. Uses the two's complement of a number as the search key
d. Divides the search list in half at each step until done
e. I Do not know

APPENDIX B

ANALYSES OF COLLEGE ENTRANCE EXAMINATION SCORES

APPENDIX B

ANALYSIS OF COLLEGE ENTRANCE EXAMINATION SCORES
FOR USE AS
PRETEST APTITUDE COVARIATE

INTRODUCTION

Before any analyses were performed of the substantive data collected as the object of this study, one-way analyses of variance were performed with Treatment as the Independent Variable and college entrance examination scores as the Dependent Variables. The purpose of these analyses was to determine whether there was any pre-treatment aptitude on the part of the participants which would be revealed as significant differences between the mean college entrance examination scores of the two Teaching Method differentiated groups.

The students who participated in the study did not all take one specific college entrance examination, and some took more than one. The distribution of the subjects by college entrance examination was the Scholastic Aptitude Test (SAT) 95, and the American College Testing Service Test (ACT) 123. Since there were only 160 participants in all, there were 58 students who took both tests.

HYPOTHESIS TESTING

The primary null hypothesis to be tested was that there will be no significant difference between the mean College Entrance Examination Composite Scores of the Guided-Discovery (Experimental) group and the Lecture-Laboratory (Control) group. Since there were two overlapping samples, two different analyses were performed initially, one for the SAT Scores and one for the ACT Scores. Finally, both the SAT Scores and the ACT Scores were converted to a common "T" scale provided by the Colorado Council on Higher Education and a third analysis was performed using this CCHET Score.

Analysis of Scholastic Aptitude Test Scores by Method

The null hypothesis to be tested by this analysis was that there would be no significant difference between the mean SAT Combined Scores of the participants when they were grouped by Teaching Method. A one-way analysis of variance was performed with SAT Combined Score as the Independent Variable and Teaching Method as the Dependent Variable. The results of this analysis revealed that there was a very highly significant difference ($p < 0.001$) between the mean SAT

Combined Scores of the two groups. The mean SAT Combined Score of the Guided-Discovery group was 1184.80, and the mean SAT Combined Score of the Lecture-Laboratory group was 1110.89. Tables B-1 and B-2 on the next page show the results of this analysis in detail.

A comparison of the mean SAT Combined scores for those students who took this test shows that the students in the Guided-Discovery Group have a mean score nearly 75 points higher than those in the Lecture-Laboratory Group. This large very highly significant difference indicates that there was a definite pre-treatment difference in favor of the volunteers in the Guided-Discovery group, at least with respect to the SAT Combined Scores.

This analysis did not support the null hypothesis that there would be no difference between the mean scores of the two groups for the SAT Combined Scores. Therefore, the null hypothesis must be rejected.

Analysis of American College Testing Scores by Method

The null hypothesis to be tested by this analysis was that there would be no significant difference between the mean ACT Composite Scores of the participants when they were grouped by Teaching Method.

TABLE B-1

SCHOLASTIC APTITUDE COMBINED SCORE
GROUP MEANS BY TEACHING METHOD

Group	Number Subjects	Mean Scores
Guided-Discovery	50	1184.80
Lecture-Laboratory	45	1110.89
Total	95	1149.79

TABLE B-2

SCHOLASTIC APTITUDE COMBINED SCORE
ANALYSIS OF VARIANCE BY
TEACHING METHOD

SOURCE	DF	SUM OF SQUARES	MEAN OF SQUARES	F	SIG-OF-F
Main Effect					
METHOD	1	129383	129383	9.96	0.00 ***
Residual	93	1207012	12978.6		
TOTAL	94	1336396	14216.98		

A one-way analysis of variance was performed with ACT Composite Score as the Independent Variable and Teaching Method as the Dependent Variable. This analysis found a significant difference ($p < 0.05$) between the mean ACT Composite Scores of the two groups. The Guided-Discovery group had a mean score of 27.53, and the Lecture-Laboratory had a mean score of 26.35. Tables B-3 and B-4 on the next page show the results of this analysis in detail.

The mean scores showed that the subjects in the Guided-Discovery Group who had taken the ACT Test had a Composite Score which was 1.18 points higher than the subjects in the Lecture-Laboratory Group. Due to a larger value for the standard deviations of the two groups, the significance of the difference is not as great as the significance of the difference between the mean SAT Combined Scores. However, the significance of the difference indicates that there was a pre-treatment difference in favor of the Guided-Discovery group, with respect to the ACT Composite Scores.

This analysis did not support the null hypothesis that there would not be any difference between the mean scores of the two groups for the ACT Composite Scores. Therefore, the null hypothesis must be rejected.

TABLE B-3

AMERICAN COLLEGE TESTING SERVICE COMPOSITE SCORE GROUP MEANS BY TEACHING METHOD

Group	Number Subjects	Mean Scores
Guided-Discovery	49	27.53
Lecture-Laboratory	74	26.35
Total	123	26.82

TABLE B-4

AMERICAN COLLEGE TESTING SERVICE COMPOSITE SCORE ANALYSIS OF VARIANCE BY TEACHING METHOD

SOURCE	DF	SUM OF SQUARES	MEAN OF SQUARES	F	SIG-OF-F	
Main Effect						
METHOD	**1**	**41.00**	**41.00**	**5.32**	**0.02**	*****
Residual	121	933.07	7.71			
TOTAL	122	974.07	7.98			

Discussion of SAT and ACT Score Analyses

Clearly, the mean college entrance examination scores for the Guided-Discovery Group versus the Lecture-Laboratory Group for both the SAT Combined Scores and the ACT Composite Scores reveal that there was an initial bias between the two groups in favor of the Guided-Discovery Group. Therefore, the use of the college entrance examination score as a covariate in the analysis of the results of the study test instrument will help to control for the initial differences between the two groups. However, not all of the students took either the ACT Test or the SAT Test and some of the students took both tests. What is needed is a method that can be used to create some common measure which can be used as the covariate.

The Colorado Council on Higher Education makes use of a conversion scale to convert ACT Composite Scores and SAT Combined Scores to a common scale identified as a T-SCORE. This common score is referred to as the CCHET in this study. The conversion scale is shown on the next page in table B-5. This table was used to convert the SAT Combined Scores and ACT Composite Scores into CCHET Scores. Four different alternatives for deriving an appropriate CCHET score were available

TABLE B-5

Chart Used for Converting ACT-Composite Scores and SAT-Combined Scores Into a Common T-Score

ACT SCORE	SAT-RANGE LOW	SAT-RANGE HIGH	COMMON T-SCORE
6	400	500	23.0
7	510	520	26.0
8	530	550	27.0
9	560	560	29.0
10	570	600	31.0
11	610	610	32.0
12	620	640	34.0
13	650	660	35.0
14	670	690	37.0
15	700	720	38.0
16	730	750	40.0
17	760	780	41.0
18	790	810	42.0
19	820	850	44.0
20	860	880	45.0
21	890	910	47.0
22	920	940	48.0
23	950	980	50.0
24	990	1020	52.0
25	1030	1070	54.0
26	1080	1110	56.0
27	1120	1150	59.0
28	1160	1200	61.0
29	1210	1250	64.0
30	1260	1310	67.0
31	1320	1350	70.0
32	1360	1430	74.0
33	1440	1470	79.0
34	1480	1490	83.0
35	1500	1600	86.0

for those students who had taken both tests. The obvious alternatives were the maximum of the two scores or the minimum, or the mean. A less obvious alternative was to use the ACT converted score since more students had taken the ACT test. It was decided that the mean CCHET Score was the most fair and unbiased.

Analysis of CCHET Scores by Teaching Method

The null hypothesis for this analysis was that there would be no significant difference between the mean CCHET Scores of the subjects when they were differentiated by Teaching Method. The SAT Combined Scores and the ACT Composite Scores were converted to CCHET Scores. A one-way analysis of variance with CCHET Score as the Dependent Variable and Teaching Method as the Independent Variable was performed. The result of this analysis was that there was a very highly significant difference ($p < 0.001$) between the mean scores of the two groups. The mean CCHET score for the Guided-Discovery Group was 61.04, and for the Lecture-Laboratory Group was 57.83. Tables B-6 and B-7 on the next page show the results of this analysis in detail.

TABLE B-6

COLORADO COUNCIL ON HIGHER EDUCATION T-SCORE GROUP MEANS BY TEACHING METHOD

Group	Number Subjects	Mean Scores
Guided-Discovery	72	61.04
Lecture-Laboratory	88	57.83
Total	160	59.27

TABLE B-7

COLORADO COUNCIL ON HIGHER EDUCATION T-SCORE ANALYSIS OF VARIANCE BY TEACHING METHOD

SOURCE	DF	SUM OF SQUARES	MEAN OF SQUARES	F	SIG-OF-F
Main Effect					
METHOD	1	408.58	408.58	10.22	0.00 ***
Residual	158	6315.82	39.97		
TOTAL	159	6724.40			

A comparison of the mean CCHET Scores shows that the students in the Guided-Discovery Group have a mean score 3.21 points higher than the students in the Lecture-Laboratory Group. The analysis of variance shows that this difference is very highly significant. Therefore, for the CCHET Score, the null hypothesis can be rejected.

In addition, the fact that the CCHET Score differences between the two groups is in the same direction as those found for the ACT and SAT Scores, and that the order of significance is similar implies that the CCHET Score is probably a viable covariate for controlling for initial aptitude differences between the two Teaching Method differentiated groups.

Table B-8 on the next page is a summary of the results for all three scores. It shows more clearly the consistancy and magnitude of the differences between the two groups. In order to more clearly show the equivalence of the three measures in this situation, the ratio of the Guided-Discovery mean to the Lecture-Laboratory mean is shown for each measure. These ratios are virtually identical for all three scoring measures which is indicative of their equivalence in this situation.

TABLE B-8

SUMMARY OF COLLEGE ENTRANCE EXAMINATION SCORE ANALYSES

SCHOLASTIC APTITUDE TEST ANALYSIS SUMMARY

			Numb	<----ANOVA--->	
Group	MEAN	STD.DEV.	Subj	F-VAL	SIG-OF-F
Guided-Disc	1184.80	120.31	50	9.97	0.00 ***
Lecture-Lab	1110.89	106.36	45		
MEANS Ratio	1.0665				

AMERICAN COLLEGE TESTING SERVICE TEST ANALYSIS SUMMARY

			Numb	<----ANOVA--->	
Group	MEAN	STD.DEV.	Subj	F-VAL	SIG-OF-F
Guided-Disc	27.53	2.61	49	5.32	0.02 *
Lecture-Lab	26.35	2.88	74		
MEANS Ratio	1.0448				

COLORADO COUNCIL ON HIGHER EDUCATION T-SCORE ANALYSIS SUMMARY

			Numb	<----ANOVA--->	
Group	MEAN	STD.DEV.	Subj	F-VAL	SIG-OF-F
Guided-Disc	61.04	6.40	72	10.22	0.00 ***
Lecture-Lab	57.83	6.26	88		
MEANS Ratio	1.0555				

TABLE B-8

SUMMARY OF
COLLEGE ENTRANCE EXAMINATION SCORE
ANALYSES

SCHOLASTIC APTITUDE TEST
ANALYSIS SUMMARY

Group	MEAN	STD.DEV.	Numb	<----ANOVA--->	
			Subj	F-VAL	SIG-OF-F
Guided-Disc	1184.80	120.31	50	9.97	0.00 ***
Lecture-Lab	1110.89	106.36	45		
MEANS Ratio	1.0665				

AMERICAN COLLEGE TESTING SERVICE TEST
ANALYSIS SUMMARY

Group	MEAN	STD.DEV.	Numb	<----ANOVA--->	
			Subj	F-VAL	SIG-OF-F
Guided-Disc	27.53	2.61	49	5.32	0.02 *
Lecture-Lab	26.35	2.88	74		
MEANS Ratio	1.0448				

COLORADO COUNCIL ON HIGHER EDUCATION T-SCORE
ANALYSIS SUMMARY

Group	MEAN	STD.DEV.	Numb	<----ANOVA--->	
			Subj	F-VAL	SIG-OF-F
Guided-Disc	61.04	6.40	72	10.22	0.00 ***
Lecture-Lab	57.83	6.26	88		
MEANS Ratio	1.0555				

A COMPARATIVE ANALYSIS OF
THE GUIDED-DISCOVERY METHOD
VERSUS
THE TRADITIONAL LECTURE-LABORATORY METHOD
IN TEACHING INTRODUCTORY COMPUTER SCIENCE

An Abstract of A Dissertation

Presented to

The Faculty of Social Sciences

University of Denver

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

---

by

Charles Paul Howerton

June 1987

The problem was to determine whether there was a method for teaching introductory computer programming which was more cost effective than the traditional lecture-laboratory technique. The subjects were entering freshmen at the Colorado School of Mines who were enrolled in either the conventional Introduction to Computer Programming class or the experimental multi-disciplinary Engineering Practices Introductory Course Sequence (EPICS) program. The subjects in the EPICS program were randomly selected volunteers.

Both groups were taught FORTRAN-77 from the same book and covered the same chapters. The method of instruction for the Introduction to Computer Programming was a traditional lecture-laboratory approach. The method of instruction for the EPICS course was guided-discovery in which the instructor provided minimal direct instruction in the form of brief introductory lectures at the beginning of laboratory oriented sessions. The EPICS instructor also provided guidelines for studying and deadlines for assignments while the students were responsible for learning the material at the established pace.

The evaluation instrument was a 40 question multiple-choice test which was administered as a post-treatment retention achievement test. The design of the test instrument was such that the questions could be partitioned into eight topical subtests to determine whether either method was better suited for presenting certain topics.

No significant differences were found by an analysis of covariance with teaching method as the independent variable, a converted college entrance examination score as the covariate and the total test score as the dependent variable. Significant differences between the two methods were found by an analysis of covariance for three of the subtests. However, similar analyses of covariance with teacher as the independent variable found that two of the three subtest differences appeared to be also dependent on the teacher of the group. The only subtest which appeared to be uniquely linked to the teaching method was "Ability to Read Programs" in which the guided-discovery group scored significantly better than the lecture-laboratory group.

Supplementary demographic analyses in which three degrees of pre-treatment exposure to computers were used as the independent variables found that almost any level of exposure provided a benefit in the form of significantly higher achievement scores.