1987

A Comparison of a Computer Programming Approach to a Textbook Approach in Teaching the Mathematics Concept "Variable" to Sixth Graders.

Enrique Ortiz
Louisiana State University and Agricultural & Mechanical College

Follow this and additional works at: https://digitalcommons.lsu.edu/gradschool_disstheses

Recommended Citation
https://digitalcommons.lsu.edu/gradschool_disstheses/4414

This Dissertation is brought to you for free and open access by the Graduate School at LSU Digital Commons. It has been accepted for inclusion in LSU Historical Dissertations and Theses by an authorized administrator of LSU Digital Commons. For more information, please contact gradetd@lsu.edu.
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.
A comparison of a computer programming approach to a textbook approach in teaching the mathematics concept "variable" to sixth graders

Ortiz, Enrique, Ed.D.
The Louisiana State University and Agricultural and Mechanical Col., 1987
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 COMPARISON OF A COMPUTER PROGRAMMING APPROACH TO A TEXTBOOK APPROACH IN TEACHING THE MATHEMATICS CONCEPT "VARIABLE" TO SIXTH GRADERS

A Dissertation

Submitted to the Graduate Faculty of the Louisiana State University and Agricultural and Mechanical College in partial fulfillment of the requirements for the degree of Doctor of Education in The Department of Curriculum and Instruction

by

Enrique Ortiz
B.A., Interamerican University of Puerto Rico, 1973
M.A., Phoenix University, 1981
August 1987

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

I wish to express my sincere appreciation to many people who have made possible the completion of this thesis. I want to thank my academic advisors Dr. L. Diane Miller and Dr. Susan K. MacGregor for taking over a big job after several committee changes. I appreciate their patience and long hours of guidance that they so willingly gave to me.

Acknowledgment must also be made to other professors who have encouraged and taken time in their busy schedule to serve in my dissertation committee. Dr. Chad D. Ellett provided wise counsel in selecting and developing the appropriate experimental design and data analysis procedures. Dr. David A. England and Dr. Rosalind N. Charlesworth gave added insight and helped me in the writing of the document. Dr. Ron Good who willingly found time to serve in the committee in a short notice.

There are others whose contributions, direct and indirect, have been most valuable to my efforts in the completion of this work. Dr. Sharon Young, while working at LSU, provided me with guidance, encouragement and educational opportunities at times when they were most needed. Dr. Mary K. Corbitt inspired and guided the early development of the study when she was working at LSU.
Finally, I want to thank my wife, Diana, for all her love, support, encouragement and patience, and my son, Enrique Gabriel, who gave me new reasons for keep trying.
TABLE OF CONTENTS

LIST OF TABLES ......................................................... viii

ABSTRACT ................................................................. xi

CHAPTER 1: INTRODUCTION ........................................ 1

Background .............................................................. 2
Rationale ................................................................. 4
Purpose of the Study ............................................... 11
Conceptual Framework ............................................. 11
Hypotheses ............................................................. 12
Treatment Effects at the End of the Study .............. 13
Treatment Effects Three Weeks after Treatment ..... 15
Treatment Effects on Attitudes ......................... 17
Supplemental Research Questions .................... 17
Relationship between Posttest and Retention Test Scores and Pre-treatment Scores .................. 18
Limitations of the Study ........................................ 20
Assumptions .......................................................... 21
Definition of Terms ................................................ 21
Summary ............................................................... 22

CHAPTER 2: REVIEW OF THE LITERATURE ............... 24

Introduction .......................................................... 24
Variables in Mathematics ........................................ 24
Defining the Concept of Variable ...................... 24
The Concept of Variable in Textbooks .................. 27
Students’ Misconceptions of the Concept of Variable .................. 30
Summary ............................................................... 37

Computer Programming and Mathematics .......... 38
Computer Programming and Mathematics Concepts and Skills ............................................. 38
Computer Programming and Mathematics
Achievement ....................................................... 52
General Mathematics ............................................. 52
Algebra ............................................................... 58
Calculus .............................................................. 65
Problem Solving and Computer Programming ....... 68
Summary ............................................................... 83
TABLE OF CONTENTS (cont.)

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning and Memory</td>
<td>85</td>
</tr>
<tr>
<td>Defining Learning</td>
<td>85</td>
</tr>
<tr>
<td>Distinction between Learning and Performance</td>
<td>86</td>
</tr>
<tr>
<td>Modern Conception of Learning</td>
<td>86</td>
</tr>
<tr>
<td>Memory and Information Storage</td>
<td>88</td>
</tr>
<tr>
<td>Sensory Register</td>
<td>89</td>
</tr>
<tr>
<td>Short-term Memory</td>
<td>89</td>
</tr>
<tr>
<td>Long-term Memory</td>
<td>90</td>
</tr>
<tr>
<td>Summary</td>
<td>91</td>
</tr>
</tbody>
</table>

CHAPTER 3: METHOD

Pre-treatment Design                                                | 93   |
Research Design                                                      | 93   |
Sample                                                               | 94   |
Instrumentation                                                       | 96   |
Pre-treatment Instructional Lessons                                  | 98   |
Treatment Instructional Lessons                                      | 100  |
Computer Programming Lessons                                          | 100  |
Textbook-based Lessons                                                | 101  |
Pre-treatment Instruments                                             | 103  |
Comprehensive Test of Basic Skills                                   | 103  |
Test of Logical Thinking                                             | 104  |
Robustness Semantic Differential                                      | 105  |
Post-treatment Instrument                                             | 107  |

Procedures                                                            | 110  |
Pre-treatment                                                         | 110  |
Treatment                                                            | 110  |

CHAPTER 4: RESULTS

Treatment Effects on Students' Understanding of the Concept of Variable | 115  |
Treatment Effects on Students' Understanding of the Concept of Variable Three Weeks After Treatment | 119  |
Treatment Effects on Students' Attitudes Toward Mathematics            | 122  |
Supplemental Analyses                                                 | 124  |
Relationship Between Students' Pre-treatment and Posttest Scores       | 124  |
Relationship Between Students' Pre-treatment and Retention Test Scores | 134  |
Intercorrelations Between Pre-treatment Measures                      | 137  |
### TABLE OF CONTENTS (cont.)

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercorrelations Between CTBS and TOLT Scores and the Post-treatment Scores</td>
<td>139</td>
</tr>
<tr>
<td>RSD Subscales</td>
<td>140</td>
</tr>
<tr>
<td>Correlations Between CTBS and TOLT Scores and RSD Subscale Scores</td>
<td>144</td>
</tr>
<tr>
<td>Multiple Regressions</td>
<td>147</td>
</tr>
<tr>
<td>Posttest Scores as the Dependent Variable</td>
<td>149</td>
</tr>
<tr>
<td>Retention Test Scores as the Dependent Variable</td>
<td>156</td>
</tr>
<tr>
<td>Reliability Analyses</td>
<td>163</td>
</tr>
<tr>
<td>Understanding of the Concept of Variable Instrument</td>
<td>163</td>
</tr>
<tr>
<td>Robustness Semantic Differential Concepts</td>
<td>164</td>
</tr>
<tr>
<td>Suggested Model to Increase R-Square</td>
<td>166</td>
</tr>
</tbody>
</table>

**CHAPTER 5: SUMMARY, DISCUSSION, AND RECOMMENDATIONS.** 170

- Effects of Treatment on Students' Understanding of the Concept of Variable at the End of the Experiment 170
- Effects on Treatment on Students' Understanding of the Concept of Variable Three Weeks after Treatment 173
- Effects of Treatment on Students' Attitudes Toward Mathematics at the End of the Treatment 176
- Relationship Between Students' Pre-treatment Scores and Their Posttest and Retention Test Scores 179
- Recommendations for Further Research 182

**REFERENCES** 189

**APPENDICES** 202

**APPENDIX A:** Pre-treatment Instructional Lessons 203
- Lesson 1 - Introducing the Turtle 203
- Lesson 2 - More Turtle World 211
- Lesson 3 - Procedures 216
- Lesson 4 - Modify Procedures and Repeat Command 221
- Lesson 5 - More Repeat Command 225

**APPENDIX B:** Computer Group Instructional Lessons 228
- Lesson 6A - Variables 228
- Lesson 7A - More Variables 231
- Lesson 8A - Still More Variables 235
TABLE OF CONTENTS (cont.)

<table>
<thead>
<tr>
<th>Lesson</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>9A</td>
<td>Variables and Repetition</td>
<td>238</td>
</tr>
<tr>
<td>10A</td>
<td>Create Procedures Using Variables</td>
<td>241</td>
</tr>
<tr>
<td></td>
<td>APPENDIX C: Textbook Group Instructional Lessons</td>
<td>243</td>
</tr>
<tr>
<td>6B</td>
<td>Introducing Variables</td>
<td>243</td>
</tr>
<tr>
<td>7B</td>
<td>Variables and Expressions</td>
<td>246</td>
</tr>
<tr>
<td>8B</td>
<td>More Variables and Expressions</td>
<td>251</td>
</tr>
<tr>
<td>9B</td>
<td>Math Machine</td>
<td>255</td>
</tr>
<tr>
<td>10B</td>
<td>More Math Machine</td>
<td>260</td>
</tr>
<tr>
<td></td>
<td>APPENDIX D: Test of Logical Thinking</td>
<td>266</td>
</tr>
<tr>
<td></td>
<td>APPENDIX E: Robustness Semantic Differential</td>
<td>276</td>
</tr>
<tr>
<td></td>
<td>APPENDIX F: Understanding of the Concept of Variable Instrument</td>
<td>278</td>
</tr>
<tr>
<td></td>
<td>APPENDIX G: Data Analyses</td>
<td>282</td>
</tr>
<tr>
<td></td>
<td>VITA</td>
<td>287</td>
</tr>
</tbody>
</table>
LIST OF TABLES

1. Summary of Pre-treatment Design ......................... 94
2. Summary of the Experimental Design ............... 95
3. Description of Sample by Treatment Group ....... 97
4. Means and Standard Deviations for Pre-treatment Measures by Treatment Group ............. 114
5. Mean Post-treatment Scores by Treatment Group . 116
6. Mean RSD Subscale Scores by Treatment Group .... 123
7. Analysis of Variance for RSD1 Scores ................. 125
8. Analysis of Variance for RSD2 Scores ................. 126
9. Analysis of Variance for RSD3 Scores ................. 127
10. Analysis of Variance for RSD4 Scores ................. 128
11. Analysis of Variance for RSD5 Scores ................. 129
12. Analysis of Variance for RSD6 Scores ................. 130
13. Pearson r Correlation Between Pre-treatment and Posttest Scores by Treatment Group .............. 132
14. Pearson r Correlation Between Pre-treatment and Retention Test Scores by Treatment Group ............ 135
15. Pearson r Intercorrelation Between Pre-treatment Scores ..................................................... 138
16. Pearson r Intercorrelation Between CTBS and TOLT Scores and Post-treatment Scores ............. 141
17. Mean RSD Subscale by Treatment Group ........... 143
18. Rotated Factor Pattern Concepts and Final Communality Estimate ............................................. 145
19. Pearson r correlation Between CTBS and TOLT and RSD Subscale Scores .................................. 146
20. Pearson r Correlation Between Posttest and Retention Test Scores and RSD Subscale Scores .... 148
LIST OF TABLES (cont.)

21. Summary of Maximum R-Square Multiple Regression of Computer Group Posttest Scores on All Independent Variables .................................................. 150

22. Summary of Maximum R-Square Multiple Regression of Textbook Group Posttest Scores on All Independent Variables .................................................. 152

23. Summary of Maximum R-Square Multiple Regression of Control Group Posttest Scores on All Independent Variables .................................................. 154

24. Summary of Maximum R-Square Multiple Regression of All Groups' Posttest Scores on All Independent Variables .................................................. 155

25. Summary of Maximum R-Square Multiple Regression of Computer Group Retention Test Scores on All Independent Variables ............................................ 157

26. Summary of Maximum R-Square Multiple Regression of Textbook Group Retention Test Scores on All Independent Variables ............................................ 159

27. Summary of Maximum R-Square Multiple Regression of Control Group Retention Test Scores on All Independent Variables ............................................ 160

28. Summary of Maximum R-Square Multiple Regression of All Groups' Retention Test Scores on All Independent Variables ............................................ 162

29. Cronbach Alpha Reliability Coefficients for the RSD Concepts .................................................. 165

30. Analysis of Variance for Suggested Model With Posttest Scores as Dependent Variable ........ 168

31. Analysis of Variance for Suggested Model With Retention Test as Dependent Variable ........ 169

Appendix Table 1. Analysis of Variance for CTBS Scores .................................................. 283

Appendix Table 2. Analysis of Variance for TOLT Scores .................................................. 284
### LIST OF TABLES (cont.)

<table>
<thead>
<tr>
<th>Appendix Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.</td>
<td>Analysis of Variance for Posttest Scores</td>
<td>285</td>
</tr>
<tr>
<td>4.</td>
<td>Analysis of Variance for Retention Test Scores</td>
<td>286</td>
</tr>
</tbody>
</table>
ABSTRACT

The purpose of this study was to investigate whether there was a significant difference in understanding the concept of variable and in attitudes toward mathematics among sixth-grade students who programmed LOGO graphics, students who used textbook-based activities, and students who received no instruction on the concept of variable.

The subjects were 89 sixth-grade students (47 female, 42 male) from two middle schools in the East Baton Rouge Parish School System. The Test of Logical Thinking (TOLT), Robustness Semantic Differential (RSD), and Comprehensive Test of Basic Skills (CTBS) were administered as pre-treatment tests.

The study was conducted in two phases, pre-treatment and treatment. The pre-treatment phase consisted of five fifty-minute lessons designed to teach all students the LOGO basic primitive commands. After these lessons were completed, the students were randomly assigned to either the computer, textbook, or control group. The treatment phase consisted of two approaches for teaching the concept of variable: computer programming and textbook-based. The computer programming approach consisted of five fifty-minute lessons in which students used variables in their graphic procedures (programs). The textbook-based
approach consisted of five fifty-minute lessons on the use of variables. The control group worked on mathematics activities during the treatment time but did not receive instruction on the concept of variable.

The data were analyzed using univariate analysis of variance procedures. The results indicated that there were no significant differences between computer and textbook groups' posttest scores, textbook and control groups' posttest scores and control and textbook groups' retention test (given three weeks after treatment) scores. Also, no significant difference was found among students' attitudes toward mathematics at the end of treatment. There were significant differences between computer and control groups' posttest scores, computer and textbook groups' retention test scores and computer and control groups' retention test scores. The significant differences favored the computer group. Also, there was a significant positive correlation between students' posttest and retention test scores and their CTBS, TOLT and computer-related RSD scores.

The results are discussed with respect to the effect of computer programming instruction on sixth-grade students' understanding of the concept of variable.
CHAPTER 1
Introduction

Today, students in our schools actively work with computers in a variety of educational settings. It has been assumed that when they program computers, they are involved in activities that promote learning of mathematics (Feurzeig, Papert, Boom, Grant & Solomon, 1969; Hatfield & Kieren, 1972; Papert, Walt, diSessa & Weir, 1979; Washburn, 1970). The theoretical link for this assumption comes from research on tutoring, learning, and practice. Camp and Marchionini (1984) stated that "in general, tutors learn from tutoring (the basis for cross-age tutoring programs), teachers learn from teaching (ask any teacher), and those who practice learn by practicing" (p.118). The question now concerns the effectiveness of computer programming activities in teaching mathematics concepts and skills. Furthermore, the use of a computer programming approach to teach mathematics compared to other approaches, such as a textbook-based approach, needs to be studied. The mathematics concept "variable" is one of the concepts that can be modeled and learned through programming activities (Corbitt, 1985). The following section presents background information regarding these ideas as they relate to the use of computer programming to enhance
students' understanding of mathematics concepts such as "variable."

**Background**

The potential positive effects computer programming could have on mathematics learning range from the enrichment of mathematical concepts to the enhancement of problem solving ability (Dprea, 1985). Most of the claims regarding the effects of computer programming on students' cognition are based on rational arguments, individual observations, and the experiences of educators and practitioners; they are not supported by systematic empirical research (Pea & Kurland, 1984).

The use of computers to enhance the mathematics teaching-learning process has received increased attention from parents, teachers, students, researchers, and curriculum developers. Reasons for this include the decreased costs and increased availability of computers for students at home and school, and the importance computers have attained in society.

Important professional groups have recommended the use of computers in the mathematics classrooms. The National Council of Teachers of Mathematics (1980) in *An Agenda for Action*, and the National Council of Supervisors of Mathematics (1977) in their "Position Paper on Basic Mathematics Skills" recommend that mathematics
programs take full advantage of computers. Moreover, the Conference Board of the Mathematical Sciences (1975) in its book titled *An Overview and Analysis of School Mathematics K-12* makes the recommendation that all students should be given the opportunity to participate in computer science courses emphasizing hands-on experiences using computers.

Computer programming provides a feasible approach for enhancing students' understanding of mathematics (Corbitt, 1985; Fey et al. 1984; Oprea, 1985). Presently, computer programming is part of the mathematics curriculum. However, Oprea (1985) states that "the objective of this instruction is the development of programming skills and not the enhancement of mathematics learning" (p. 2). It is necessary and important to combine computer programming capabilities with effective teaching. Transfer of computer programming skills to noncomputer activities is not made automatically by students (Kurland & Pea, 1984). Determining effective ways of using computer programming to enhance students' understanding of mathematics and studying the effect of computer programming on students' understanding of specific concepts, such as "variable" and "equation," are basic if educators want to be successful in using this approach in the mathematics classroom.
Rationale

The concept of variable is considered a very important unifying theme in mathematics. It is one of the few central themes which serve to organize, unify, and give coherence to mathematics at all levels. It is therefore important that teachers and curriculum developers analyze the effectiveness of different approaches in teaching of the concept of variable.

Concepts are considered to be one of the foundations of knowledge (Sowder, 1980). Through learning certain concepts, other concepts and other kinds of subject matter are learned (Cooney, Davis & Henderson, 1975). The concept of variable is no exception. Wagner (1981) states that "Just as the concept of number is fundamental to arithmetic, the concept of variable is fundamental to algebra and higher mathematics" (p. 107). Fey et al. (1984) feel strongly that the importance of the concept of variable in mathematics becomes more relevant because the secondary school mathematics program for college-bound students begins with, and is dominated by, topics from algebra.

Variables are essential to mathematics in many ways. They are used in mathematical expressions, sentences, formulas, and functions. They can serve as a blank, hold a place for a constant, represent an unknown in the solution of conditional equations, describe a family of
functions or graphs (where they are called parameters, or arbitrary constants), or denote cross-referenced symbols such as in the study of logic (Karuch, 1962). Educators might assume that because the concept of variable is so important to mathematics there is sufficient study on the teaching and learning of this concept. This, however, is not the case as very little research exists in the area, despite evidence that variable is a troublesome concept for many students and that students have several misconceptions regarding the use of this concept. For example, Carpenter et al. (1981) found that 13-year-old students could solve simple linear equations, but they did not appear to understand formal procedures for finding solutions to more complex equations, and Tonnessen (1980) found a low level of acquisition of the concept of variable at the college level. Also, Wagner's (1981) findings suggest that middle and high school students have several misconceptions concerning the concept of variable. The investigator stated that these misconceptions center on a lack of understanding of the algebraic processes involved and in the students' confusion with the different uses of the concept. However, the concept of variable has not received the attention it deserves in textbooks and in research studies. Surprisingly enough, it is not even introduced in several elementary school textbooks.
Computer programming provides a feasible approach to teach and help students avoid or overcome misconceptions of the concept of variable. In computer programming, as in mathematics, a variable is a symbol that may be replaced by any member of a given set of numbers. Each variable corresponds to a distinct location in the "memory" of a computer. This idea makes computer programming involving variables a powerful and concrete model to introduce and teach the concept of variable (Papert, 1980). The development of sound field testing and conscientious selection of activities using computer programming to teach the concept of variable are essential. If teachers use the technology available correctly, then students could be positively influenced by it (Fey et al., 1984). Corbitt (1985) recommends that computers should be used to facilitate an early introduction of algebraic concepts such as variable in the elementary school.

The effectiveness of computer programming activities to enhance the quality of mathematics instruction, and more specifically to teach the concept of variable, might be influenced by several factors. Researchers need to explore computer programming approaches to teaching the concept of variable and how well they facilitate students' understanding of the concept. There are several learning theories that might be used as a conceptual framework for
studying the effectiveness of computer programming on teaching the concept of variable. Bloom's (1976) theory of Learning-for-Mastery is one of these theories. Bloom suggests that by matching the opportunity to learn (time on task) and the quality of instruction, teachers should be able to ensure mastery by the student. In this regard, computer programming can improve the quality of instruction by providing a concrete model and a direct application of the concept that a textbook-based approach can not match.

Bloom (1976) also states that if the opportunity to learn and appropriate activities are used to teach mathematics, three factors should account for most of the variation among students in achievement. The variables identified by Bloom are cognitive entry behavior (the extent to which the student has mastered prerequisite knowledge and skills), affective entry behavior (the extent to which the student is willing to engage actively in the learning process), and the quality of instruction (the extent to which instruction is appropriate to the student). The prerequisite knowledge and skills needed in a computer task can be kept minimal by using a simple and practical computer language. In general, computer programming activities provide for the affective entry. Students are willing to participate in computer activities and find them motivational.
There is a need to analyze the impact of new technologies and approaches to the teaching of mathematics concepts such as "variable." Moreover, there is a need to compare the effectiveness of programming as an instructional approach with traditional textbook-based exercises for teaching mathematics concepts. LOGO is a procedural and graphical computer programming language available for use on microcomputers. A procedure that the computer can store to be used at a later time is defined as a group of one or more instructions to the computer. LOGO makes it possible for beginners to learn to program in a relatively short period of time.

Also, this computer language allows learners to order a "turtle" to move forward or backward and turn left or right; for example, the command FORWARD 20 will tell the turtle to move forward 20 turtle steps. The turtle is a triangular figure displayed on the computer screen in the draw mode of LOGO. In this setting, variables can be used in programs to make geometric shapes of any given size. For example, students can write, run, and edit a program that makes squares of any given size.

The study of how LOGO graphics instruction affects students' understanding of the concept of variable has not been studied thoroughly. Several studies have suggested that students can enhance their understanding of mathematical concepts through computer programming.
Clement, Lockhead & Soloway (1982) with fifth graders using BASIC, Oprea (1985) with sixth graders also using BASIC, and Milner (1972) with college students using LOGO have found a positive correlation between programming ability and understanding of the concept of variable. The results of the present study would reinforce previous findings and generalize the theory to a larger population using another computer programming language.

Computer programming might not only provide an effective model to learn mathematics, but also it might help students retain what they have learned (Cruikshank, Fitzgerald & Jensen, 1980). Students' retention of their understanding of the concept of variable could be enhanced by the concrete experiences provided by computer programming activities involving variables. Moreover, how students learn a concept or skill might be an important factor when trying to remember what they have learned (Cruikshank, Fitzgerald & Jensen, 1980).

Most researchers in this area have concentrated their efforts on investigations of general transfer of programming skills to mathematics. Transfer of programming skills to other nonprogramming activities such as problem solving, general mathematics, algebra and calculus has been found difficult to accomplish even for adult programmers (Lecuyer, 1977; Robitaille, Sherrill & Kaufman, 1977; Ronan, 1971). Studies of novice and
expert programmers have shown that the claimed transfer of programming skills to nonprogramming activities might not occur unless students have acquired a high level of proficiency in computer programming. Therefore, students might need more time than what is given in order to develop the required programming expertise.

Analyzing students' gains in understanding the concept of variable through computer programming activities instead of analyzing students' general gains in mathematics achievement is an example of the "micro" transfer approach. The micro transfer approach has been found useful in research given the difficulty of studying macro transfer effects. A look at transfer of specific programming skills has been suggested as a more appropriate approach to analyze transfer of programming skills to nonprogramming activities. The approach focuses on single mathematics concepts learned through programming and isolates specific factors that may contribute to the development of more general abilities (Billings, 1983; Hungate, 1982; Swigger & Campbell, 1981). The study of novice programmers' transfer of variable perception through programming activities might give some insight on how transfer of specific concepts occurs, and what programming expertise students need to make such transfer.
**Purpose of Study**

The purpose of this research was to investigate whether there was a significant difference in understanding the concept of variable and in attitudes toward mathematics among sixth grade students who used LOGO graphics (computer group), students who used non-computer activities (textbook group), and students who received no instruction to learn the concept of variable (control group). In addition, the strength of the relationship between students' posttest and retention test scores and their pre-treatment scores was analyzed.

**Conceptual Framework**

Bloom's (1976) theory of Learning-for-Mastery provides a useful conceptual framework to analyze the instructional effectiveness of a textbook-based approach and a computer programming approach to teach the concept of variable to sixth grade students, and to depict the cognitive and affective variables included in the study.

Bloom (1976) considers students' cognitive and affective entry levels as important factors affecting learning outcomes. According to Bloom (1976), students enter each new learning task with a particular history of previous development and learning. This previous history will determine the nature of the students' interaction with the learning activities and learning outcomes. The
The hypotheses of this study are presented in three sections. The first section presents hypotheses related to treatment effects on students’ understanding of the concept of variable at the end of treatment. The second section presents hypotheses related to treatment effects on students’ understanding of the concept of variable three weeks after treatment. The final section presents a hypothesis related to treatment effect on students’ attitudes toward mathematics tested at the end of the treatment.
Hypothesis 1: Students receiving computer programming instruction will demonstrate greater understanding of the concept of variable on the posttest than students receiving textbook-based instruction.

It was expected that students receiving programming instruction would have greater understanding of the concept of variable than students receiving textbook-based instruction and students receiving no instruction (control group students) on the concept of variable. In Bloom's (1976) terms, computer programming would provide for improving the quality of instruction. LOGO computer programming presents concrete props that could provide for sixth grade students' understanding of the concept of variable. These concrete props are visual representations of the effect of changing the value of a variable in a LOGO procedure that the students can manipulate. The students will be able to see how changing the value of a variable changes the design on the computer screen. This provides a thinking model that might help students understand and apply the concept of variable in different situations. This approach would be more effective than the textbook approach which lacks concrete props as the ones presented by the LOGO graphics. Some sixth grade students are capable of formal reasoning but still require concrete experiences in order to learn concepts (Inhelder
and Piaget, 1958). These students profit more from learning activities that incorporate concrete and real world experiences based on students' conceptual orientation (Piaget, 1976). Computer programming could provide sixth grade students the proper experiences which could enhance their involvement in the activities (time on task) and understanding of the concept of variable.

Hypothesis 2: Students receiving computer programming instruction will demonstrate greater understanding of the concept of variable on the posttest than students receiving no instruction on the concept of variable (control group students).

It was expected that students receiving computer programming instruction would have greater understanding of the concept of variable than students receiving no instruction. In other words, it was expected that some instruction would be more effective than no instruction at all. Also, this provides control for possible effects due to maturation and history during the treatment. The students might have a greater understanding of the concept of variable even without having the treatment.

Hypothesis 3: Students receiving textbook-based instruction will demonstrate greater understanding of the concept of variable on the posttest than students receiving no instruction on the concept of variable (control group students).
As in the case of the hypothesis above, it was expected that some instruction would be more effective than no instruction at all. If the textbook approach were more effective than no instruction on the concept of variable, then students receiving textbook-based instruction should demonstrate greater understanding on the posttest.

**Treatment Effects Three Weeks after Treatment**

**Hypothesis 4:** Students receiving programming instruction will demonstrate greater understanding of the concept of variable on the retention test (three weeks after treatment) than students receiving textbook-based instruction.

It was expected that students receiving computer programming instruction would not only outperform students receiving textbook-based instruction at the end of treatment but also would outperform students receiving textbook-based instruction three weeks after treatment. It was suspected that students' understanding of the concept of variable through programming activities was more permanent.
Hypothesis 5: Students receiving programming instruction will demonstrate greater understanding of the concept of variable on the retention test (three weeks after treatment) than students receiving no instruction on the concept of variable (control group students).

As with the hypothesis dealing with students' posttest scores, it was expected that computer programming instruction was more effective than no instruction on the concept of variable. This also provides control for possible effects due to maturation or history.

Hypothesis 6: Students receiving textbook-based instruction will demonstrate greater understanding of the concept of variable on the retention test (three weeks after treatment) than students receiving no instruction on the concept of variable (control group students).

It was expected that students receiving textbook-based instruction would not only have greater understanding of the concept of variable but also would retain their understanding of the concept of variable three weeks after treatment. The quality of instruction and time on task were considered more effective in the textbook approach than with no instruction on the concept of variable.
Treatment Effects on Attitudes

Hypothesis 7: Students receiving programming instruction will have more positive attitudes toward mathematics at the end of the treatment than students receiving textbook-based instruction or no instruction on the concept of variable.

It was expected that students receiving programming instruction would have more positive attitudes toward mathematics at the end of the treatment because of the motivating effects computers have on students. The computer programming effect on some components of school mathematics is primarily affective (Papert, 1980). Papert (1980) suggests that "many children have come to the LOGO lab hating numbers as alien objects and have left loving them" (p. 68).

Supplemental Research Questions

The following research questions were analyzed using data from the semantic differential affective scale, and test of achievement, logical thinking, and understanding of the mathematics concept "variable" (posttest and retention test). Bloom (1976) stated that if the opportunity to learn and appropriate activities are used to teach mathematics, three variables should account for most of the variation among students' gains in understanding. The variables identified by Bloom are
cognitive entry behavior, affective entry behavior and quality of instruction. Two variables were included as indicators of students' cognitive entry behavior, mathematics achievement and logical thinking, and affective entry behavior.

**Relationship between Posttest and Retention Test Scores and Pre-treatment Scores**

Two supplemental questions were related to students' scores in the affective scale. They were the following: (1) What is the relationship between students' posttest scores by treatment group and their scores in the affective scale?; (2) What is the relationship between students' retention test scores by treatment group and their scores on the affective scale? Students' affective entry was considered as one of the variables that would explain some of the variation among students' learning of the concept of variable just after treatment and three weeks after treatment. It was expected that students' affective entry behavior would have a significant positive relationship to students' gains in understanding of the concept of variable at the end of the treatment and three weeks after treatment.

The next two supplemental questions were related to students' mathematics achievement: (1) What is the relationship between students' posttest scores by
treatment group and their scores in the mathematics achievement test? (2) What is the relationship between students' retention test scores by treatment group and their scores in the mathematics achievement test? Mathematics achievement was one of the cognitive entry behaviors considered in the study. It was expected that mathematics achievement should account for some of the variation among students' gains in understanding of the concept of variable at the end of the treatment and three weeks after treatment. A significant positive relationship could imply that mathematics achievement was a significant factor in explaining variation in students' performance and it should be taken into account when considering the introduction of programming activities involving variables to sixth grade students.

The final two supplemental questions examined were related to students' logical thinking: (1) What is the relationship between students' posttest scores by treatment group and their scores in the logical thinking test?; (2) What is the relationship between students' retention test scores by treatment group and their scores in the logical thinking test? Another cognitive entry behavior considered in the study was logical thinking. Sixth grade students are in a transition from concrete operational to formal operational thought (Inhelder and Piaget, 1958). These students are capable of formal
reasoning, but they can still profit from concrete experiences. The computer programming approach provides a concrete model from which students can learn the concept of variable. Work with the turtle provides specific intuitive models for complex mathematical concepts most children find difficult, as is the case of the concept of variable for sixth grade students (Papert, 1980). The understanding of the concept of variable is facilitated by using LOGO computer programming (Papert, 1980). A significant positive relationship could imply that some logical thinking is required from the students in order for them to learn the concept of variable through programming activities.

Limitations of the Study

1. Generalizations are limited to sixth grade students from the Baton Rouge, Louisiana, metropolitan area with no or very limited experience in computer programming in LOGO or any other computer language.

2. The programming instruction is limited to features included in the turtle graphic capabilities.

3. While the experimental unit was the student and the activities were designed to allow for independent work, uncontrolled teacher effect on groups' performance might have accounted for some of the variations among students' understanding of the concept of variable at the
end of the treatment.

Assumptions

1. Random assignment of the students to the different groups will provide homogeneous groups.

2. The programming instruction, if any, students receive prior to the study has a limited effect on students' computer programming ability and mathematics understanding.

3. Prior learning of the concept "variable" will have a limited effect on students' understanding of the concept.

Definition of Terms

1. **Computer Programming Instruction** - Instruction will focus on the teaching of LOGO graphics commands and the concept of variable in a guided discovery approach. The teacher will guide students' understanding of the commands, helping them to overcome possible misconceptions that may arise when working through the lessons.

2. **Variable** - A symbol used to represent any value of a given set.

3. **Domain** - The set of all possible values that the variable can assume.

4. **LOGO** - A procedural computer language. One of the features of LOGO is "turtle graphics." The display
creature in the turtle graphics is a triangular figure known as the "turtle." The turtle resides on the computer screen in the draw mode of LOGO. The turtle graphics allow the learners to order the turtle to move forward and backward and turn left or right. The user can have the turtle leave tracks (lines) on the screen when it moves, or the turtle can be told to move without leaving tracks.

5. Understanding of the Concept of Variable - Students' understanding of the concept of variable will be measured with a test devised by the investigator.

6. Textbook-based Approach - Instruction will focus on the teaching the concept of variable in a noncomputer environment. The teacher role is to guide students' learning of the concept of variable and to overcome the students' misconceptions.

Summary

Computer programming provides a concrete embodiment of mathematics concepts such as "variable," or at least a concrete visualization, and activities that make student learning more meaningful (Hart, 1981). There is a common pedagogical intuition among educators that programming activities help students' facility in the use of mathematical concepts and skills and in their understanding of the way particular mathematical concepts operate (Hart, 1981). Some of the claims have been
investigated through research, but there is inconclusive evidence of the effects of computer programming on students' mathematical development. The few controlled research studies conducted in this area give only general direction concerning the possible cognitive benefits of programming. The proposed research study is essential if teachers and educators want to use computers successfully in the classroom to enhance students' learning of mathematics. Clements and Gullo (1984) state that "computers will soon be an integral part of the classroom and home environment of children, yet there are unanswered questions concerning their effect on young children's cognition" (p. 1051). Also, attention should be given to students' characteristics.

Bloom's (1976) theory of Learning-for-Mastery was used as a theoretical framework. Students' cognitive entry (mathematics achievement and logical thinking) and affective entry behaviors were considered as possible factors explaining variations among students' gains in understanding of the concept of variable at the end of the treatment and three weeks after treatment. The following chapter reviews the literature that relates to the concept of variable as it is used in mathematics, the use of computer programming to enhance students' understanding of mathematics, and the relationship between memory and learning.
CHAPTER 2

Review of the Literature

Introduction
The central premise of this review is that computer programming has some cognitive effect on students' understanding of mathematics. This chapter reviews literature that relates to the concept of variable, the use of computer programming to enhance students' understanding of mathematics, and the relationship between learning and memory.

Variables in Mathematics
This section reviews research that relates to the concept of variable in mathematics. The focus will be on defining the concept of variable, the concept of variable in textbooks, and students' misconceptions of the concept of variable.

Defining the Concept of Variable
Defining the concept of variable is a difficult task. Collis (1975) identified the concept of variable as one of the ways literal symbols can be interpreted in algebra. His ideas have helped to refine the term "variable" and to understand the ways students use literal symbols. He described six ways students use literal symbols: letter
evaluated, letter ignored (not used), letter as object, letter as specific unknown, letter as generalized number, and letter as variable. The first three are ways children may interpret letters to avoid the formal theoretical understanding implicit in a given task. The other three are different theoretical uses of letters in algebra and require a higher level of abstraction.

Variables have also been defined as placeholders. This idea has been used to argue that literal numbers are essentially just like verbal expressions and that mathematics is a language in which variables are an essential part (Beberman, 1964; Gupta, 1980; Hockett, 1967; The language of mathematics, grades 6-12, 1981; Russell, 1940; Tarsky, 1941). In this context, variables are called pro-numerals to make an analogy with the pronouns in the English language. Wagner (1983), however, indicates that placeholder is an important characteristic of literal symbols, but that it overstates the similarity between letters and words, and understates the power of literal symbols.

Other definitions can be found in several books and dictionaries (Gibson, 1981; Glenn, 1943; Marks, 1966; Young, 1916). These sources give adequate definitions of the concept of variable and provide examples of this concept. In general, the authors define "variable" as a
letter or symbol that may designate or represent any value in a given or specified set of numbers.

Tonnessen (1980) states that a valid definition of the concept of variable must incorporate the concepts of symbol and domain, and that these concepts are connected by the relational concept "represents any element of." The investigator indicates that the defining attribute "is used to represent an arbitrary element of a given set" and is one of the critical attributes of the concept of variable. The following five irrelevant attributes were also identified: kind of variable, whether the symbol is labeled "constant;" whether, for a given equation, an arbitrary element of the domain is a solution; and whether any of the elements of the domain, when substituted, result in an undefined expression.

Tonnessen's definition incorporates the concepts of symbol and domain without identifying the concept of variable as either:

A variable is an ordered pair \((x, D(x))\) where \(x\) is a symbol and \(D(x)\) is a set with at least two elements, such that \(x\) represents any of the elements of the set \(D(x)\). The set \(D(x)\) is called the domain of \(x\), and elements of \(D(x)\) are called the values of \(x\) (p. 10).

This definition describes variable as an ordered pair and makes explicit the interconnectedness of domain and symbol. It is not consistent with the conventional practices of labeling the symbol as "variable." No
research evidence was found regarding the educational utility of this definition. The following section examines the definitions given to the concept of variable in elementary and secondary school textbooks.

The Concept of Variable in Textbooks

Since the present investigation is more concerned with the teaching of the concept of variable as it is taught in elementary and secondary schools, definitions from textbooks at these levels are most relevant. Four categories of textbooks were reviewed: 1) elementary school textbooks (from first grade to eighth grade) (Eicholz et al., 1987; Nichols et al., 1985; Payne et al., 1985); 2) general mathematics (Berardi, Jones & Foster, 1982; Gerardi, Jones & Foster, 1983; Keedy et al., 1983); 3) algebra one and pre-algebra (Coxford & Payne, 1983a; Foster, Rath & Winters, 1983a; Nichols et al., 1980; Pearson, Duffy & McCaffery, 1984a; Price, Rath & Leschensky, 1982); 4) algebra and trigonometry (Coxford & Payne, 1983b; Foster, Rath & Winters, 1983b; Johnson et al., 1982; Keedy, Bittinger and Smith, 1982; Nichols, Edwards & Henry, 1982; Pearson, Duffy & McCaffey, 1984b).

In elementary school textbooks, the concept of variable is indirectly presented in books one through six and introduced somewhat formally later in books seven and
eight. In books one through six, the concept of variable is used as a placeholder or blank to develop the idea of missing value in addition, subtraction, multiplication, and division. For example, $2 + \square = 11$. This idea is used basically to demonstrate the relationship between addition and subtraction, and multiplication and division. It is also used to help students recall fact families. Literal symbols are not used in these textbooks.

The use of variables in this context does not help students learn the relational concept "represents any element of." The students "merely review the arithmetic facts stored in their memory; therefore, their guessing of the answer amounts to mental computation" (Herscovics & Kieren, 1980, p. 573). These solutions, which are based on numerical values, cannot be described as algebraic processes. This is evidenced by the fact that the same students who solve this type of exercise correctly cannot solve similar equations when large numbers are involved (Herscovics & Kieren, 1980).

Usually, the concept of variable is introduced somewhat formally in books seven and eight of the elementary school textbooks. A definition of the concept of variable followed by examples are sometimes given. As in the definitions examined in the previous section, the concept of variable was defined as a symbol that may represent any number in a given set. Another type of
exercise usually presented is to evaluate expressions for a given variable; for example, in the expression $X + 4 = 12$, what is the value of $X$?

In the textbooks related to general mathematics, algebra one, pre-algebra, algebra, and trigonometry, a similar definition was given together with examples and exercises. Also, the concept is used in algebraic equations, functions, and formulas. Nonexamples, and explanations of why an example is an example were not given in these textbooks. A definition identifies the necessary and sufficient conditions, and the examples clarify them. However, if reasons are given for the examples, these reinforce the necessary or sufficient conditions (Cooney, Davis & Henderson, 1975).

According to Cooney, Davis and Henderson (1975) the definitions given to the concept of variable in these textbooks are arbitrary and elegant moves since they employ a minimum of language, but the very elegance of these definitions may be a block to learning. Some students might be confused by using this type of definition, or might not understand the logic and essence of the definition. Students' misconceptions of the concept of variable could be related to how they interpret these definitions. The use of instructional models to teach the concept might help students understand the
algebraic processes involved and avoid misleading conceptions.

**Students' Misconceptions of the Concept of Variable**

Most of the research in the area of the concept of variable is related to identifying students' misconceptions in understanding this concept. Carpenter et al. (1981) reported results of the National Assessment of Educational Progress. The instrument used contained 85 exercises intended to measure students' ability to work with the concepts of variable and relationship. Most of the exercises dealt with algebraic concepts and manipulations, and with basic variable concepts. The following categories were included: variables in equations and inequalities, variables used to represent elements of a number system, functions and formulas, and coordinate systems. Thirteen- and seventeen-year-old students were included in the sample. The seventeen-year-old students' performance was also analyzed based on mathematics course background.

The investigators found that most of the 13-year-old students could solve simple linear equations intuitively, but they did not appear to know formal procedures for finding solutions to more complex equations. Fewer than 50 percent of the 17-year-old students with a year of algebra could systematically solve linear equations; 66
percent of those with two years of algebra could do this task effectively. The investigators argued that this difference could be related to the additional instruction the students received; generally, the most capable students are the ones who take a second course in algebra. The investigators also found that performance on algebra problems was consistently low, and the format of the problem significantly affected performance. Students appeared to be more familiar with using a box rather than a letter to represent a variable. Seventeen-year-old students' performance was found to be related to course background.

Tonessen (1980) conducted a study to measure the levels of attainment by college mathematics students of the concept of variable. Eight criteria were formulated to assess the attainment of the concept of variable, four corresponding to concept acquisition and four to concept use. Acquisition of the concept was determined by the following criterion tasks: concept identification, formulation of the concept, and positioning the concept of variable within a hierarchy of concepts. Concept use was determined by the following criterion tasks: determining the equivalence of variable expressions, substituting the same value for a variable each place the variable occurs in a given discussion, classifying expressions as phrases or as open or closed sentences, and determining when two
expressions are related by a change of variable. The level of mathematics considered to develop these criterion tasks was pre-calculus college algebra.

The criterion tasks were used to assess students' attainment of a correct or ungeneralized concept of variable. A refined conception of the concept of variable refers to the knowledge a person has when he/she understands the critical attribute of the concept to be critical and the irrelevant attributes to be irrelevant. An ungeneralized conception of variable is the knowledge a person has when he/she understands the critical attribute to be critical but also considers one or more of the five irrelevant attributes to be critical to the concept.

The sample was composed of the following subgroups: 202 algebra students, 178 trigonometry students, and 186 calculus students from the University of Wisconsin. Two equivalent forms of a paper and pencil test were administered to these students. The reliability coefficients (Kuder Richardson-20) calculated for each form of the test ranged from .53 to .71. These coefficients were considered low by the investigator. The content validity was considered high.

The investigator found a low level of correct responses on concept acquisition items, whether the standard was a correct or an ungeneralized conception of the concept of variable. The mean level of correct
responses on these items was considered very low when compared to performance on concept use items. This was true for the three populations assessed. This finding was considered by the investigator as evidence that it was not necessary to have a correct or ungeneralized conception of variable in order to use it effectively. There was no statistical evidence to sustain that the level of concept acquisition can be used to predict the levels of concept use. The findings of this study are limited by the fact that the assessment instrument had a low reliability and that it was developed for a particular college level population.

Clement, Lockhead and Monk (1981) studied the translation difficulties in learning mathematics that are relevant to analyze students' understanding of the concept of variable. The written questions included in this study are based on a prior series of interviews in which college science students were asked to talk aloud while working on a simple word problem named The Students and Professors Problem. It reads as follows:

Write an equation, using the variables S and P to represent the following statement: "At this university there are 6 times as many students as professors." Use S for the number of students and P for the number of professors (p. 286).

In these interviews, fewer than 50 percent of the students could solve the problem presented in the interview. The predominant error was reversing the variable in an
equation, \( 6S=P \) rather than \( 6P=S \).

The sample used in the study consisted of 150 calculus level students and 47 non-science majors taking college algebra. These students were tested using a written test with word problems similar to the one above. Also, 15 students were asked to think aloud while solving the word problems. The investigators found that "the reversal error appears to be due not simply to carelessness but rather to a self-generated, stable, and persistent misconception concerning the meaning of variables and equations" (Clement et al., 1971, p. 289). They argued that students cannot overcome these misconceptions by doing more manipulation of equations; this technique usually does not require students to understand the meaning of an equation and the concept of variable.

Rosnick (1981) investigated students' misconceptions of the concept of variable. A two part multiple choice test based on The Students and Professors Problem above was administered to 33 sophomore and junior business majors in a statistics course. Most of these students had had two semesters of calculus. The version of the students and professors problem reads: At this university, there are six times as many students as professors. This fact is represented by the equation \( S=6P \). Part A asked the following question: In this
equation, what does the letter P stand for? Part B asked the following question: What does the letter S stand for? The investigator found that over 40 percent of the students were incapable of selecting "number of professors" as the only appropriate answer for Part A. These students used letters to represent a less abstract idea, "professor" or "professors."

Over 43 percent of the students failed to answer part B correctly. Also, over 22 percent of the students chose "S stands for professor" as their answer for part B. These same students chose "none of the above" as their answer for part A. Rosnick argued that these findings support "the hypothesis that students tend to view the use of literal symbols in equations as labels that refer to concrete entities" (p. 412), and the previous finding that the reversal error is not a common one but is one that is self-generated, stable, and persistent.

Wagner (1981) presented a unique idea concerning the concept of variable. The objectives of her clinical research were to "a) illustrate one way of extending Piaget's conservation methodology from simple concepts to related concepts, and b) investigate students' ability to conserve equation under alphabetic transformation of literal variables" (p.107). Conservation tasks were used to assess students' understanding of equation and variable by determining whether or not a student realizes that the
essence of the concept (critical attribute) is invariant under transformations of certain properties (irrelevant attributes) that seem to be related to the concept but are not. Clinical interviews were conducted with thirty students, half of middle school and half of high school level. One conservation-of-equation task and three conservation-of-function tasks were developed and administered to the students.

The investigator found no significant associations between age group and ability to conserve equation or function. Some students seemed to think that changing the literal symbol in an equation task may also change the solution of the equation. These students confused the linear order of the alphabet with the linear order of whole numbers thinking, for example, that the expression 3W+5 will give a result smaller than the expression 3X+3 because W comes before X in the alphabet. The small size and arbitrary nature of the sample were suggested as limitations of the study. Also, only one task of each type was used as a criterion measure which might not be enough to decide if a student is a conserver, nonconserver or transitional conserver.
Summary

Research findings suggest that students have several misconceptions concerning the concept of variable. These misconceptions are centered on a lack of understanding of the algebraic processes involved and in students' confusion with the different uses of the concept. The investigators tend to agree that teachers have an important role in helping students avoid these misconceptions. Appropriate approaches to teach the concept of variable at various levels are necessary in order to be effective in this task. This is related to Bloom's (1976) ideas of the quality of instruction. Bloom states that "the instructional variable of greatest importance is believed to be the Quality of Instruction, the extent to which the cues, practice, and reinforcement of the learning are appropriate to the needs of the learner" (p. 11).
This section reviews research that relates to the use of computer programming to enhance students' understanding of mathematics. The following areas are included: computer programming used to enhance students' development of general mathematical concepts and skills, computer programming used to improve students' mathematics achievement, and computer programming and students' problem solving ability.

The following discussion analyzes studies that have examined the idea of using computer programming to enhance students' understanding of mathematical concepts and skills.

In a comparative study, Washburn (1970) coordinated computer programming exercises with mathematical topics of a particular course at junior high, twelfth-grade, and college freshman levels. He tested the claims that students' programming skills can be used to strengthen their understanding of mathematical concepts and that this approach can be successfully used with students who differ greatly in characteristics (age, intelligence, and level of mathematical achievement).

The students in the experimental group received instruction in the programming language APL and sets of
programming exercises. They wrote, executed, and corrected computer programs as part of the treatment. The students in the control group were required to perform homework assignments of a conventional nature not related to computer programming. The exercises were developed to relate to the mathematical topics the students encountered in their mathematical course. Also, the students were required to re-examine their knowledge and understanding of the concepts and skills included in the course.

A pre- and posttest involving the mathematical topics covered in the study were administered to both the experimental and the control groups. Also, a testing instrument was used to measure students' attitudes toward the computer-enriched mathematics program approach. The investigator concluded that the approach used in the study can strengthen the students' understanding of mathematical concepts. This gain was found to be independent of students' age and level of mathematical achievement.

Several studies have been developed using LOGO programming as a conceptual framework for teaching mathematics. Feurzeig et al. (1969) investigated the teaching of mathematics in terms of LOGO computer programming. They also explored means of using this computer language as a foundation for the mathematics curriculum. The investigators suggest that teaching with a suitable programming language can contribute to
mathematics instruction in several ways. One of these ways is that programming can be used to give students very "specific" insight into a number of important mathematical concepts, for example, the concepts of variable and function. In their report the investigators included two studies.

The first study involved second- and third-grade students who were taught LOGO programming in a non-controlled educational environment. The investigators assumed that given this environment the students would develop "formal reasoning" at a much earlier age. The investigators found that these students learned the "elements" of LOGO programming with ease, but some of them could not learn to write or "debug" complex computer programs during a short period of time. They observed that the students acquired a meaningful understanding of concepts like variables, functions and formal procedures (though not those words) through LOGO programming. The investigators based their findings on non-quantitative or informal observations of students working with the computers.

The second study involved 12 seventh-grade students (six boys and six girls) of "average" ability. These students were randomly selected from a population of 100 seventh-grade students. They received a treatment consisting of one hour class discussion and a half hour in
computer programming for one semester. Most of the time they worked independently, but sometimes they worked in pairs. The goal of the study was to give the students an introduction to high school algebra which they normally would not have studied before the ninth grade. The investigators concluded from their informal observations that it is feasible to teach LOGO programming and effectively teach mathematics using this computer language as a conceptual and operational framework.

It is important to note that the LOGO version used in these two studies is an early version of the language which did not have the "turtle" graphic capabilities and other features that more recent versions have. Also, the studies are informal in nature and, as such, have several uncontrolled variables, such as students' age and time on task. They provide valuable and interesting information, but more research is needed to test their findings and claims.

In another study using LOGO programming, Papert et al. (1979) conducted a project with 16 sixth-grade students. The report is presented through the analysis of anecdotal material. The study is based on the belief that students learn mathematics and thinking skills through computer programming in a goal-directed environment. This is a more extensive study that includes several objectives. One of the objectives involves learning the
elements of LOGO computer programming which includes the following ideas and skills:

1. Computer instruction as a formal language (including the language syntax, effect and associated error messages).

2. Sequential procedures and the ability to translate an informally defined plan into a working program.

3. Use of sub-procedures and superprocedures.

4. Editing and debugging programs.

5. Control of continuing process with loops and/or recursion.

6. Use of variables, conditionals and stop rules.

7. Writing interactive programs.

The data collection was based on records of each student's interaction with the computer, the teachers' anecdotal records of each student's daily work, regular and occasional observations conducted by the MIT LOGO group, and a series of pre/post student interviews.

The investigators reported that of the 16 students, two did not learn to program in a significant sense. Students' entry achievement was considered the main factor affecting these students' performance; both of them were "low" achievers (as measured by standardized test CTBS). Other students ("high" achievers) were able to develop "understanding" of more advanced concepts, such as
variables. They found that most of the students developed a concept of "micro-world" which was constructed by the students to explore and become more comfortable with the computer. Though the idea of command was not easily understood by the students, the idea of procedure was found to be a significant and valuable one. Also, they suggested that the use of variables in LOGO programs to change the size or shape of a geometric figure provided a concrete introduction to the use of variables and that students' understanding of the concept of variable in computer procedures might be a major step in the development of mathematical thinking or higher levels of abstraction.

Lawler (1980) reported an observation and analysis of his eight-year old son working with several LOGO procedures involving variables and recursion. This child had participated in other studies before, including an intensive six-months study at the LOGO Project of the MIT Artificial Intelligence Laboratory (Lawler, 1979). The child had, as observed by the author, two mental predispositions that he brought to his work. He was inclined to call upon symmetry as a generative idea, and he approached problems in a surprisingly systematic way for one his age.

The investigator found that the child applied his past experience with LOGO to solve problems involving
patterns. Also, he found that using LOGO in a goal-directed exploration aided the child's understanding of the concept of variable.

Given the characteristics of the child studied and the fact that only one student was considered, it is difficult to generalize the study's findings. The major threat to the external validity of the study is the multiple-treatment interference because multiple treatments have been applied to the same respondent (Campbell & Stanley, 1963). In this case, we do not know the amount of time the student has spent with the computer and the extra attention he has received in his previous experience.

Some studies in the area of using computer programming to enhance students' understanding of concepts have used BASIC computer programming. Dwyer (1975) presents a preliminary report regarding another quantitative study based on the belief that student-controlled programming activities will enhance students' mathematics learning. The project was a follow-up of a three year program in the Pittsburgh public school system. The students used five labs in a noncontrolled learning environment. The labs were also a way of organizing the mathematics content as they were designated as a computer lab (focusing on algorithms), a dynamics lab (focusing on processes that take place in
time), a logical design lab (digital and analog circuit models), a synthesis lab (focusing on using superimposition and synthetic music), and a model of reality lab (focusing on modeling and manipulating concrete materials).

The investigator found that students' directed exploration of mathematical concepts such as variables in a computer programming environment fostered their learning of those concepts. This was determined by direct observation of students' work with LOGO. This report could have been improved by including additional information regarding the description of sample and procedures used in the study. Without this information it is difficult to make any generalization of the findings and makes the findings inconclusive.

In another study using BASIC programming, Oprea (1985) investigated the effects of computer programming instruction on students' understanding of variables and mathematical generalizations. Two different instructional strategies were used in this study, wholistic and elemental. The wholistic approach began instruction at the whole program level and the student learned computer programming through the programs themselves. The other instructional strategy focused on the elemental commands of the programming language and proceeded stepwise until the students learned computer programming with complex
problems.

Two sixth-grade classes at each of two different elementary schools were used as treatment groups. One class was randomly assigned to each of the two instructional approaches (wholistic and elemental). The students received computer programming instruction during 60-90 minute sessions two or three times a week for six weeks. Pre- and posttests assessing programming ability, mathematical generalization skills, and understanding of the concept of variable were administered to the students. The items on the understanding of the concept of variable instrument were adopted from other studies. Using the posttest scores of all students, Hoyt's Reliability estimate of Cronbach's alpha was calculated. The Variable Instrument had a reliability of only .67, which was considered low. Factor analysis was used to determine the content validity of the instruments. The results of the factor analysis supported the validity of the instruments. These results were considered as the most significant; the investigator argues that "it is validity which measures the relationship of the data obtained to the purpose for which it was collected" (Oprea, 1985, p. 67).

The investigator concluded that sixth-grade students can learn computer programming and that this learning enhanced their understanding of the concept of variable and mathematical generalization. However, she was unable
to provide evidence for the claim that different instructional methods would influence students' computer programming learning.

Krull (1980) conducted a study with elementary school students, also using BASIC, designed to engage students in writing computer programs involving five properties in mathematics and to measure the effects of computer programming on achievement and attitudes toward school. The author analyzed the effects of learning to program a computer on students' performance in mathematics.

The experimental group was taught BASIC programming and spent several weeks developing and writing computer programs involving specific mathematical properties. The control group was enrolled in a remedial mathematics program. A pre- and posttest for mathematics achievement and an attitude toward mathematics survey were administered to both groups in order to test the effectiveness of the program. The investigator found that programming activities had a significant positive effect on students' learning of mathematics. Also, he found that there were no significant differences, with respect to attitudes, between the experimental and the control groups.

One major concern regarding the validity of the findings is that the control and experimental groups did not receive comparable instruction. The investigator did
not explain the content, instructional style, or teaching strategy used with the control group. Also, sample selection and assignment of students to control or experimental groups were not discussed by the investigator. The students might have differed in important ways. These limitations make the findings hard to interpret and generalize.

Frichard (1985) supported the claim that mathematical concepts and skills could be introduced and developed through computer programming activities that involve algorithms. Mathematics iteration is considered an important area in the development of an algorithmic approach in mathematics. It is defined as "a process whereby a sequence of terms are produced in such a way that \( a_n \) is determined from its predecessors in the same way that \( a_{n-1} \) is determined from its predecessors" (pp. 1-2). The investigator examined the development of student's knowledge of iteration as they worked with computer programming. She did not report the computer language used in the study. Four high school students were involved in computer programming activities involving construction, execution, revision and refinement of computer programs. A tenth-grader and a ninth-grader (who were in the same geometry class and did not have previous experience with computing), a tenth-grader (who was enrolled in a second-year algebra and had some limited...
experience with computers), and a ninth-grader (who was also taking geometry and had some experience with computing) participated in the study.

These students received a ten week treatment that consisted of 13 group teaching sessions (held twice a week), and three individual interviews for each student. Five sessions dealt with problems and situations related to the generation of numerical sequences and eight sessions with iterative methods for solving quadratic equations. The students worked on assigned programming tasks in pairs (one pair per computer).

The data consisted of records of students' work, video-tapes, and audio-tapes of interviews and teaching sessions. Three components of the students' knowledge of iteration were analyzed. The first component was the specific initial value (or values) of the variables which were used in the procedure. The investigator found that the students were aware that by changing a starting value they would also change the sequence produced by the program.

The second component analyzed was the repeated procedure. This component is necessary but not sufficient to produce an iteration. In this area the students developed programs to solve problems and to develop arithmetic and geometric sequences. Programs were suggested and refined by the students and teacher as a
group, then students typed and executed the programs. The third component was the stopping rule which presented several problems to the students. It required understanding of iteration and the role of the stopping rule.

The four students displayed different levels of understanding regarding the different components of iteration. Two students were able to describe the component of the concept, and the other two demonstrated their understanding as they worked through the tasks. The investigator concluded that computer programming provided the context within which the students shaped their understanding of iteration, the generation of sequences, and the solving of equations.

Although the above studies found that computer programming seems to enhance student’s understanding of mathematics skills and concepts, other studies have produced conflicting results concerning this idea (Johnson, 1971). Hoffman (1971) designed a quasi-experiment to study the effect of computer extended instruction on generalization skills and achievement in a second year algebra course. Two instructors at two metropolitan area schools taught both an experimental and a control class. The control groups were taught in a traditional manner without computer applications, while the experimental groups were taught with computer
applications. The experimental groups used BASIC programming and programming problems in addition to eight computer extended units. The investigator administered a standard pretest and two posttests (a generalization skills test related to functions and the same standardized test administered as the pretest) to the four groups.

The researcher found no evidence to support the hypothesis that computer extended instruction significantly affected generalization skills or achievement based on total posttest scores. However, he found that the experimental groups were successful with certain generalization skills. It is claimed that these skills seem closely related to the process followed when the student "debugs" or refines a malfunctioning program.

The possibilities for generalization of the study's findings could have been enhanced by using a random sample of students. Also, more descriptive data are needed on the type of students and teachers participating in this study. There is no information regarding teacher selection and assignment of groups to teachers. Also, there is no information regarding time on task. Another concern is that the test used to examine students' generalization skills was not described in any way. No assessment of the reliability and validity of this test was provided in this report.

According to the research examined to this point,
computer programming appears to enhance students' understanding of mathematics concepts and skills, such as variables, iteration, functions, and procedures. Also, the investigators have found several misconceptions, limitations, and difficulties that students have when learning programming which might diminish computer programming effectiveness, and make research findings inconclusive. This could be related to some of the findings of research with novice and expert programmers that will be examined later in this dissertation.

**Computer Programming and Mathematics Achievement**

The studies examined in this section investigate the idea that computer programming experiences enhance student's mathematics achievement in the areas of general mathematics algebra and calculus. Some of these studies also examine specific concepts and skills through their investigations.

**General Mathematics.** In an experiment using programming as a conceptual basis for learning mathematics, Milner (1972) investigated the effects of teaching computer programming on performance in mathematics. The investigator hypothesized that the students would learn mathematics through programming activities including the learning of the concept of
variable. Students in the experimental (18) and students in the control (20) groups were randomly selected to participate in the fifteen-week study from a population of fifth grade students at the school. The computer group received a three-phase treatment, five weeks for each one (two sessions each week). The phases were as follows: learning of the elements of LOGO programming, training in the writing of computer programs, and completing tasks that dealt with the generation of arithmetic and geometric sequences (no information other than task definition was given to the students). The LOGO language used in this study is an early version of the language.

A test was designed by the investigator and several mathematics educators, and administered (both as pre- and posttest) to examine students' understanding of the concept of variable in mathematics before and after the experiment. To develop the instrument, the investigator first began with defining the concept of variable as used in mathematics. After that, behavioral objectives were formed based on the definition and translated into test items. The reliability of the instrument using pretest and posttest correlation for the noncomputer group was found to be .77. Analysis of covariance was used to test for statistical differences among experimental and control groups. The three treatments used were as follows: incomplete computer program, algorithm given in natural
language, and no information given other than task definition.

The investigator concluded that computer programming was effective and of value in students' mathematics learning. Performance on the test showed significant differences between students' competency in using variables favoring the experimental group. Also, the programming students demonstrated their competency in using variables by explaining their written computer programs. The particular instructional approach was found less important than the description and development of tasks suitable for the students.

More control was needed for instructional differences between the experimental and control groups. There was no specific or supplementary treatment described for the control group. The differences between the experimental and control groups might have been caused by uncontrolled variables such as time spent at the computers and special attention given to the experimental group.

Howe, O'Shea and Plane (1979), researchers who were highly trained programmers, spent two years teaching LOGO programming to eleven eleven-year-old boys of "average" or "below" average mathematics ability. These students spent one year learning computer programming and one year learning mathematics through LOGO programming. Each student worked one hour per week in a computer programming
classroom.

The nonprogrammers had pretest scores significantly better on the Basic Mathematics Tests but equivalent pretest scores on the Mathematics Attainment Test. Comparisons between programmers and nonprogrammers showed that programmers had improved enough in mathematics to eliminate the original differences (Basic Mathematics Test scores) with nonprogrammers but had significantly lower scores on the Mathematics Attainment Test. These global differences do not support claims that computer programming provides some justification for, and illustration of, formal mathematics rigor (Pea & Kurland, 1984).

The investigators found that the LOGO group learned to talk sensibly about mathematics issues and concepts, and explained mathematics difficulties clearly. They observed that using LOGO supports the idea of a child's self-directed exploration of mathematical concepts such as variables on computers. These findings are based on differences in ratings between programmers and nonprogrammers on teacher questionnaires. They are unreliable because teachers knew beforehand which students belonged to which group; they should have been unaware of the students' placements to make their ratings reliable (Pea & Kurland, 1984).

Kieren (1969) found mixed results in a comparative
study of the use and nonuse of computers. The study was designed to examine if the use of computer programming would affect achievement in mathematics at the high school level for students of "average" previous achievement (based on their scores in a standardized mathematics achievement test). The study took place over a two year period with 36 students. The computer language used was not reported.

Jordan (1985) found similar results with seventh-grade students in a similar study. The experimental group used BASIC programming language to review some mathematics skills and concepts. The control group used the traditional method (textbook, paper and pencil) for mathematics drill and practice. The same teacher taught both groups.

The investigator found no significant differences between the mean gain for the control and experimental groups on the California Achievement Test subsections of mathematical computation, concepts and problems. Students in the control group did learn mathematics. However, students in the experimental group had increased two skills, computer programming and mathematics.

Hatfield and Kieren (1972) studied the effects of computer programming at grades seven and eleven for two years in a general mathematics course. They hypothesized that students of "higher" ability probably prefer to deal
with mathematical principles and problems in the traditional deductive way, whereas students of "lower" ability benefit from the organization required in computer programming. The students were randomly assigned to experimental and control groups, and blocked for analysis purposes into three levels of previous achievement performance. The topics and methodology were kept the same for both groups, except that the computer group wrote and processed computer programs in BASIC involving the concepts, problems, and skills discussed in class.

Both standardized (Coop Structure of the Numeration System, Contemporary Mathematics Test, Step Mathematics 3B, and Iowa Test of Basic Skills) and teacher-made tests were administered to the students. Analysis of variance and analysis of covariance were used as statistical methods to compare main effects and differential effects of treatment across previous achievement levels. Comparisons of proportion of students responding correctly to a test item were made to analyze the particular effects of the treatment.

The investigators concluded that eleventh-grade students of "higher" ability appeared to benefit less from computer programming experiences than less capable students. While the evidence was not conclusive, computer programming appeared to have some facilitating effect on mathematics achievement. The ongoing development of
materials in this study might have affected its findings. Also, there is a concern regarding the reliability and validity of some of the instruments used in the study.

**Algebra.** In algebra, also, inconsistent results have also been found in research relating computer programming and mathematics achievement. Lamb (1977) found positive results favoring the use of computer programming to improve students' achievement. The investigator compared the achievement of algebra students who wrote, processed, and evaluated computer programs dealing with discrete combinatorial algorithms with achievement of those who did not use computers to learn the same topics. The purpose of the study was to examine the effects of treatment, ability, and pretesting on achievement in graph theory. Ninety students were categorized as "low" or "high" ability groups according to previous performance in mathematics and employing an extended Solomon four-group design. These students were randomly assigned into 8 different groups.

The students were pretested by using instruments developed by the investigator (Graph Theory Achievement Test and Rating Revised View of Mathematics Inventory). The data were analyzed by using a three-way analysis of variance to determine main and interaction effects of treatment, ability, and pretesting. Also, the Spearman
Rho and the Kendall Tau tests were used to assess correlations and associations involving the instruments used.

The investigator found that the main effect of the treatment factor on achievement in graph theory content and the main effect of the ability factor were significantly related to achievement, but the effect of pretesting factor was not significant. No evidence was found in terms of change in students' attitudes toward mathematics.

The researcher concluded that this study provided evidence to support the feasibility of coordinating the study of combinatorial algorithms with computer programming in the classroom; in particular, students with access to a computer scored significantly higher on the tests assessing these topics than students who did not use this approach. The programming language used was not identified in this report which is an important aspect of the study. The instruments used were described in some detail, but neither an assessment nor an analysis of reliability and validity were included.

Other studies have found mixed results or no significant differences between the use and nonuse of computer programming in teaching algebra. Robitaille, Sherrill and Kaufman (1977) evaluated three ninth-grade algebra classes in each of two schools and collected and
analyzed data concerning students' attitudes and achievement. The investigators used a computer-augmented approach to teach the experimental groups.

In each school, one class made use of the computer in learning algebra throughout the treatment period. A second class made use of the computer for the first third of the treatment period and was given no further access to the computer for the remainder of the treatment. The control group had no exposure to the computers. The students were randomly assigned to the different groups. All the classes in one school were taught by the same teacher. In the other school, the teacher effect was partially controlled because the computer and partial-computer groups were taught by one teacher and the noncomputer group by another. The three classes studied the same mathematics content using a contemporary algebra textbook. The students in the computer and partial-computer groups used BASIC to write and run computer programs on topics related to their algebra classes. Standardized tests were administered to measure students' attitudes toward mathematics and their verbal and mathematical abilities before and after the experiment.

The results of this study did not support the use of computer programming in algebra. The noncomputer group did better in achievement than the computer group. In the
area of attitudes toward mathematics, a significant difference was found favoring the computer in a short-term evaluation, but in the long-term evaluation there were no significant differences between the groups.

Factors like absenteeism and change of class or school reduced the number of subjects initially included in the study. These are factors affecting the validity of the results. Experimental mortality of respondents from the comparison groups is considered an extraneous variable relevant to the internal validity of the study (Campbell & Stanley, 1963).

In an experiment that lasted one nineteen-week semester, Ronan (1971) also found mixed results with high school students. Students, classified as a "middle" ability group based on their prior achievement in mathematics, in two algebra-trigonometry classes participated in the study. The students were randomly assigned to the control and experimental groups taught by the same teacher except for instruction and assignments in the experimental group involving the use of the computer and the BASIC language which were given by the investigator. The course objectives, methods, techniques, and instructional materials were the same for both groups except for the use of the computer by the experimental group as a computational tool and as a teaching-learning tool.
The investigator found that the students using the computer to learn mathematics achieved significantly higher than students who did not use computers to learn mathematics in exponential functions and logarithms, significantly lower in trigonometric identities and formulas, and no differently radicals in equations, complex numbers, and circular functions. Also, students who used the computers to learn mathematics achieved significantly higher in tests of logic and reasoning than students who did not use computers. These results are similar and closely parallel the Hatfield and Kieren (1972) results in the area of general mathematics achievement of seventh-grade students.

Saunders (1979), similarly to Hoffman (1971) in the area of mathematical concepts and skills development through computer programming, noted that the use of computer programming had no significant effects on achievement or attitudes of algebra students. Also, the investigator found that the introduction of this approach into the traditional algebra curriculum does involve some uncertainty for "high" ability students. In this study, computer-enhanced resource materials were used without taking time away from instruction in algebra topics. Comparisons of the data obtained from an experimental group and a control group were made to determine the effects of using this approach in teaching algebra. The
experimental group worked assignments in addition to regular lessons. The control group did not use these additional assignments. Four groups were used for comparisons, two experimental groups and two control groups.

Another investigation (Lecuyer, 1977) had two major components: a textbook for college students using a programming language and an exploratory study into the use of APL programming language in teaching an undergraduate survey course in mathematics. The study was conducted at the University of Massachusetts for two class sections. One section was taught using APL and the computer. The other section was taught the same topics, and used the same textbook but without APL or the computer.

The experimental and control groups answered a multiple-choice diagnostic pretest developed by the investigator and designed to measure students' understanding of the topics covered in the study (Set Theory, Functions and Graphing, Exponentials and Logarithmic Functions, Statistics, Vectors and Matrices, and Systems of Linear Equations).

Lecuyer (1977) found no significant differences in achievement (as measured by a standardized test) between the two groups at the end of the experiment. He concluded, however, that students in the experimental group had as good an understanding of the mathematical
concepts as those students who learned in the conventional way. This implies that the computer programming students learned two skills in the same amount time—computer programming and mathematics.

Katz (1971) directed a research study in which average second-year algebra students studied in three different ways. One experimental group wrote computer programs in conjunction with regular classroom presentation of 19 different algebra topics. Students' programs were run by computer aides and then returned to them. In a second experimental group, the students also wrote computer programs in conjunction with learning algebra, but they ran their own programs on the computer. Time spent in the computer room was taken from the regular classroom time. A control group receiving instruction using traditional procedures learned the same topics. The students were randomly assigned to the different groups. The computer facilities consisted of a Digital Equipment Corporation PDP-8/S computer, along with one line and one off-line teletype.

The Cooperative Mathematics Test-Algebra II was administered to the students. Also, two tests developed for the study (consisting only of algebra topics that were related to computer programming) were administered as mid-term and final examinations. Comparisons made among treatments based on mid-term and final examinations.
revealed no significant differences among the three groups. The most effective use of the computer was having students write computer programs that were run by aides. The students running their own programs did not appear to benefit from the experience.

**Calculus.** Similarly to Milner (1972) in the area of general mathematics achievement, the following two studies investigated the effects of computer programming on achievement in introductory calculus courses. The first one, by Bitter and Slauchert (1971), is an experiment in which the experimental and the control groups (from each of three participating colleges) received the same mathematical content by the same instructor. In the experimental group, the normal assignments were replaced by computer homework exercises. They used programmed materials during the first week to learn BASIC programming and after this first week had to write their own programs to solve the assignments.

The investigators found that the students who received computer-oriented instruction achieved significantly higher than those who did not receive the treatment. No significant differences were reported in the integral calculus test items.

The second study was developed by Holbien (1971). The research was conducted during the fall quarter, 1969,
at Morehead State College. Treatment utilized in the morning classes was replicated in the afternoon (29 students in the morning and 30 students in the afternoon). Students were randomly assigned to classes at each time period. Two teachers were involved, each one teaching an experimental and a control group.

The investigator required the use of a computer terminal to complete homework assignments during a portion of class time (about 50%) for the experimental groups. The students were required to write computer programs to obtain numerical results to mathematical concepts being developed. Homework assignments were the same for the experimental and control groups, except for the computer programs assigned to the experimental groups.

Teacher-made tests were used to assess students' achievement in, and attitudes toward mathematics. Analysis of variance was conducted to analyze the data. The investigator found that there was evidence indicating that achievement for one of the experimental groups was significantly higher than for its corresponding control group. Also, there was evidence that students of lower ability were helped more by this approach of limited access to the computer terminal than those students of higher ability.

Statistical analysis of the proportion of correct responses to an item by the different groups showed that
some items were handled better by students in the experimental groups. Two concepts predominated after categorizing the items answered correctly more frequently by a treatment group. They were the items dealing with the limit of function at a point, and evaluating a function. No single concept or concept area was found to be answered correctly by more control group students.

In a similar study, Bell (1970) conducted a six-week experiment at Cornell University to determine the effectiveness of teaching an introductory calculus course using a computer-oriented approach. The course topics included elementary functions, probability, and calculus. The control groups (46 students) studied calculus by using a calculus manual written by the investigator. The experimental group (49 students) studied calculus by using a similar manual, except that it included six computer oriented programs. T-test and analysis of covariance were used for testing the hypotheses, using means of scores obtained on four calculus examinations (SAT-mathematics score was used as covariate).

Similarly to Holoien (1971), Bell found significant differences favoring the computer group. He concluded that a computer oriented approach to calculus is an effective way to promote students' understanding of concepts and to increase students' interest in calculus. He also claims that this approach does not interfere with
students' learning to apply techniques of calculus.

Problem Solving and Computer Programming

It is difficult to make final conclusions concerning the effectiveness of computer programming augmented problem solving in mathematics classrooms based on research findings. The following is a review of the literature related to this area.

One of the early claims of the effectiveness of computer programming comes from Feurzeig et al. (1969). They claim that computer programming activities provide a context for problem solving and language with which the student may describe his/her own problem solving reasoning. Several studies have been developed around this idea, but still there is not enough conclusive evidence.

Papert (1973) argued in favor of this claim, noting that writing programs of LOGO "turtle" geometry are new pieces of material that allow clear discussion and simple models of heuristics. Also, in Mindstorms, Papert (1980) discusses a pedagogy surrounding LOGO, and suggests the cognitive benefits that will emerge from taking "powerful ideas," such as recursion and variable, in "mind-size" bites (procedures). He sees "no reason to doubt that this difference could account for a gap of 5 years or more between the ages at which conservation of numbers and
combinatorial abilities are acquired" (p. 175). This claim is based on extensively replicated findings of a large gap between early conservation of numbers, near age 7, and later combinatorial abilities, near age 12 (Pea & Kurland, 1984).

In a study already examined in this report, Papert et al. (1979) reported, through anecdotal material, that children engage in extensive problem solving and planning activities when learning programming. Whether or not such activities had cognitive effects beyond programming was not studied. However, Evans (1985) studied the use of LOGO computer programming as an exploratory environment to facilitate natural learning and problem solving. The areas identified for investigation were the following: problem solving, motivation to learn, and attitudes toward learning mathematics. Schools preselected by a school system program with fourth-graders were involved in the one semester study. The basis for selection was not indicated in this report. A non-equivalent control group design was used to control for this difficulty. Four standardized tests were used for all fourth-grade students as pretest and posttest. One way analysis of covariance was performed on each of the four measures. The pretest scores were used as a covariate. The investigator concluded that LOGO programming had positive effects on students' problem solving abilities and attitudes toward
learning mathematics.

Selection bias is one of the major concerns in this study. Also, there is a need for more description of the population used, the number of schools participating, the differences among these schools, and the number of students participating in the study. Teacher effect is another important uncontrolled variable affecting the validity of the findings.

Howe et al. (1979) investigated the computer programming effects on 11 through 13-year-old boys, divided into two groups of eleven students each. The experimental group, of lower initial ability compared to other students, went to a LOGO computer classroom during normal school hours, and the control group followed the normal schedule. The students' performance on various tests was recorded several times during the two-year study.

The experimental group spent the first year learning computer programming. Ideas such as variables, procedures and recursion were introduced to the students. Also, problem solving strategies like decomposition, and the use of debugging skills were discussed with the students. The second year the students used computer programming to explore mathematics topics (arithmetic, algebra and geometry).

Howe et al. (1979) concluded that the experimental
group improved in their ability to do mathematics and in their understanding of mathematics as compared to the performance of the control group. The classroom teachers were of the opinion that the students in the experimental group "could argue sensibly about mathematics issues and explain mathematics difficulties clearly" but rated the control group poorly for this. These findings suggest that computer programming can benefit the less able student. Unfortunately, uncontrolled factors might have accounted for the differences between the control and experimental groups. The extra personal attention given to the students in the experimental group and the extra time spent working on mathematics are examples of these uncontrolled variables.

Clements and Gullo (1984) investigated the effects of learning computer programming compared to experiences in computer-assisted instruction (CAI). The study focused on six-year-old children's cognitive style (reflective and divergent thinking), metacognitive ability, cognitive development (operational competence and general cognitive measures), and ability to describe directions at the end of the twelve-week treatment. Eighteen first-grade children who were randomly assigned to one of two treatment groups (computer programming or CAI) participated in the study. The students were pretested to assess receptive vocabulary, impulsivity/reflectivity, and
divergent-thinking abilities. The following instruments and tasks were administered: Peabody Picture Vocabulary Test-Revised (PPVT-R), Matching Familiar Figures Test (MFFT), Torrance Tests of Creative Thinking-Figural Test, two tasks to evaluate cognitive processes (metacognition), four tasks on logical operations (classification and seriation), four subtests of the McCarthy Screening Test, and a describing directions task.

The investigators found significant differences on the Torrance Tests of Creative Thinking in favor of the computer programming group on fluency, originality and overall divergent thinking score. No significant differences were found in favor of the CAI group. Also, there were significant differences on the MFFT for the LOGO group on error and latency. The LOGO group significantly outperformed the CAI group on the metacognition tasks (ability to monitor and evaluate their own cognitive processes). No differences were found between the groups in cognitive development.

Clements (1986) designed a research project to investigate delayed effects of computer programming in LOGO on mathematics and cognitive skills in primary grade children. To assess the long-range effects of LOGO programming, children who participated in a study developed by Clements and Gullo (1984) were tested and interviewed in their third grade year (18 months after the
end of the training). Only two of the eighteen students who participated in the study by Clements and Gullo (1984) did not participate in the study developed by Clements (1986), both students from the CAI group. The instruments used in this study were the following: the Peabody Picture Vocabulary Test-Revised (PPVT-R); Test of Cognitive Skills (TCS); and California Achievement Test, Level 13 (CAT, L 13).

Clements (1986) found that LOGO programming affects children’s ability to complete items that demand the application of metacomponents. The metacomponents are executive skills or processes that are utilized in planning and evaluating one’s information processing. These skills or processes include: deciding on the nature of the problem, selecting a mental representation, deciding on performance components relevant for the solution of the problem and on a strategy for combining those components, and monitoring solution processes. Also, there was evidence that LOGO had a diffused and delayed effect on certain areas of achievement and problem solving. The investigator suggested the possible lack of connections between the schemata children may develop in LOGO and those they develop in the mathematics classroom. He indicated that if LOGO is used to develop mathematical competencies, then “mappings” between childrens’ computer experiences and mathematics experiences must be made and
brought to a level of explicit awareness for the children. Given this mapping, the development of mathematical competencies is possible through computer programming, but what is not known is how efficiently this can be done. The investigator suggested that "future studies might utilize LOGO training in which teachers assist children to form correct and complete mathematical concepts, and to construct mappings between the LOGO experience and their regular mathematics work" (Clements, 1986, p. 10).

In general, these two studies support previous findings regarding the positive effects of programming on some aspects of students' problem-solving ability and understanding of mathematical concepts. These studies provide evidence that computer programming may affect students' cognitive style, but there is no evidence that it affects general cognitive development. The findings, however, are limited by the small size of the sample.

In a study discussed in the previous section, Bell (1970) developed a six-week experiment to determine the effectiveness of teaching introductory college calculus using a computer-oriented approach. The control group (46 students) studied calculus using a calculus manual. The experimental group (49 students) studied calculus using a similar manual containing six problems to be solved by using computer programming. The investigator found that students in the experimental group did more and better
mathematical problem solving than those who did not write
computer programs. This finding is limited by the
informal nature of the study.

Some studies have found no significant effects, and
others have found mixed effects when examining the use of
computer programming to enhance student’s problem solving
ability. Collenbanck (1983) investigated the relationship
between problem solving performance of high school
trigonometry students and the degree of their exposure to
computer programming. The 98 students selected for the
study were enrolled in one of four regular or enriched
trigonometry classes in a public high school (2300 total
enrollment).

One regular and one enriched class which participated
in BASIC computer programming served as experimental
groups. The control group received the same instructional
experiences, except computer programming experiences. A
pre- and posttest of problem solving ability developed by
the instructor and a computer programming test to
ascertain subjects' awareness of computer programming
processes were administered to the students. Analysis of
variance, analysis of covariance, and descriptive
statistics were used to analyze the data. The
investigator found no evidence to support the conjecture
that the computer group outperformed those with no
computer experience in problem solving situations. Also,
no evidence was found to support the relationship between one's problem solving performance and his/her computer programming knowledge.

Rose (1983) found mixed results regarding the use of computer programming to enhance students' problem solving ability. Forty fifth-grade students were paired by matching composite scores on the Iowa Test of Basic Skills (ITBS), followed by random assignment to experimental and control groups. Gains in logic were measured by using the Cornell Critical Thinking Test (CCTT), and gains in subject-oriented problem solving were measured by using the Developing Cognitive Abilities Test (DCAT). Both tests were administered as pretest and posttest. The experimental group received about 20 hours of programming experience, working in pairs through a set of printed lessons. The computer language used in this study was not identified in this report. Analysis of covariance was used for statistical tests.

The investigator found no significant differences between the experimental and control groups on the DACT scores, but the experimental group showed significant gains on the CCTT and on one of its subsets. The researcher concluded that computer programming instruction appeared to improve the problem solving abilities of the students, especially in the areas of hypothesis testing and determination of relevant data, and suggested a
possible transfer of computer programming skills to a more
genral problem solving ability. However, academic
problem solving was not improved by the short-term
instruction provided in this study.

Foster (1972) investigated students' problem solving
performance with regard to non-routine problem experiences
using computer programming and flow charts as tools. The
investigator hypothesized that students with experience
using these tools in problem solving situations would
outperform students without these experiences. A Problem
Solving Ability Test (PSAT) was administered to measure
nine selected behaviors associated with problem solving:
specifying conditions a datum satisfies, selecting a
relevant solution, posing a hypothesis, identifying a
pattern, supplying missing information, selecting relevant
data, using a constructed algorithm, correcting an error
in a constructed algorithm, and constructing an algorithm.

Sixty-eight eighth-grade students from a suburban
public school were partitioned into four treatment groups:
no supplementary aid, flow charts only, computer only, and
both computer and flow charts. Twenty-four nonstandard
problem tasks (12 required and 12 optionals) allowing for
both computer and noncomputer solutions were provided to
each student over a 12 week period. The treatment groups
received supplementary aids and training on how to use
them. A four by three factorial design was used which
included the factors of treatment and class, and reading nested within each treatment. The PSAT (assessing the areas identified above) was administered as a posttest but only to assess students' performance. Analysis of variance and multivariate analysis of variance were used to analyze the mean performance of each group on this test. Pairwise comparisons of mean performance of the four treatment groups were made by using a one-tailed Dunnett t-statistic.

Foster (1972) presented evidence that computer programming supports the development of problem solving behaviors. The mean performance of the treatment group with computer only was significantly higher than that of the group with no supplementary aids. He concluded that "computer programming (and to a lesser extent flow charting) tends to support the development of selected problem solving behaviors, especially within verbal problem solving experiences" (p. 4239).

The evidence presented to this point is not conclusive, but the balance of the evidence is favorable with respect to the use of computer programming to enhance students' problem solving ability (Kieren, 1973). However, these studies do not present evidence regarding the claim that computer programming activities promotes learning of more general problem solving skills beyond programming. The following two studies investigated the
supposition that problem solving skills learned through computer programming may prove useful or may have cognitive effects beyond programming.

Statz (1974) carried out a study to assess this supposition. LOGO programming was taught to a group of 16 students, ranging from nine to eleven years of age. Before and after the treatment, the students were tested in four problem solving tasks. A group of students of the same age level and ability participated as the control group. The concepts of procedure, recursion, variable, conditionals and editing were examined in a four-stage acquisition framework. These concepts were searched for in other contexts to study the effect of the programming experience.

The investigator found that in two (word puzzle and a permutation task) of the four tasks the computer group did significantly better than the noncomputer group. In the other two tasks (Tower of Hanoi task and a horse problem developed by the researcher) no significant differences were found. The experimental group scored significantly higher than the noncomputer group on a test of recursion. The hypothesis that children who have had LOGO experience for several semesters will perform significantly better on problem solving than students who did not have this experience was neither confirmed nor rejected. More research in this area was suggested by the investigator.
Also, the data showed a tendency to support the claim that students who worked with LOGO conditionals will do better in tests of problem solving. However, it failed to support the hypotheses that the problem solving test scores of children in the computer group are highly correlated with the level of LOGO learning, and that the computer group will have more positive attitudes toward creative problem solving than students in the noncomputer group.

The investigator interpreted these findings as mixed support to the claims that learning LOGO programming promotes the development of more general skills. Success was reported with transfer of a specific concept, recursion.

In the other study, Bruggeman (1986) used four experimental groups and one control group to examine the above supposition and also to examine the effects of four instructional methods upon solving computer programming problems. One-hundred and thirteen sixth-graders enrolled in a computer programming course participated in the study.

The investigator found that participation in the computer programming course had no effect upon participants' ability to solve non-routine mathematical word problems, but significant associations were found with the Iowa Problem Solving Test used as pretest and the
computer programming posttest. Training with less complex programs was found to improve the students' debugging skills, but neither the complexity nor the instructional methods gave a relative advantage in solving more complex programs. Similarly to Statz (1974), Bruggeman concluded that these findings support the idea which favors the teaching and study of domain-specific rather than general problem solving skills. The following study focused on this idea.

Soloway, Lochhead and Clement (1982) tried to isolate specific factors contributing to more general results, focusing on simple problems. The study was based on the findings of another study by Clement, Lochhead and Monk (1981) in which they found that students seemed to have considerable difficulty in translating simple algebra word problems from written description of the problem into an algebraic equation. Analysis of error patterns suggested that a conceptual misunderstanding might have been related to students' responses. Most of the errors were reversal errors: 6S=P instead of S=6P and 4C=5S instead of 5C=4S. They argue that these errors occur because the students are adopting a descriptive rather than a procedural approach, and that programming provides an environment in which students can develop a procedural view of problem solving.

A pretest/posttest design was used to examine
students' performance on algebra word problems similar to the ones used in the above mentioned study. Subjects were given one equivalent form of the test at the beginning of the semester and the other form at the end of the semester. The students participating in the study were enrolled in an introductory programming course using Pascal. A control group was selected matching the experimental group background as closely as possible. There were 132 students participating in the experiment.

The investigator found that more students solved the problems correctly when they were expressed using a computer programming approach rather than an algebraic approach. They attributed this finding to the procedural semantics of equations in programming that might be missed in the algebraic task.

The computer programming effect found in the above study is much more restricted in scope than the increments in general problem solving skill presented by the cognitive transfer claims (Pea and Kurland, 1984). But as suggested before, it could be a good approach to analyze the effects which learning computer programming has on students' cognition. This becomes more important because we are not certain whether intellectual activity is guided by general domain-independent problem solving skills or by domain-dependent problem solving skills (Pea and Kurland, 1984). Ross and Howe (1981) suggest that in "most
problem solving tasks it is impossible to apply the supposed context-free skill without initially having essentially domain-specific knowledge" (p. 21).

Summary

The research examined so far regarding the effects of computer programming on students' cognitive development is not conclusive. However, there is evidence that there are, in fact, some cognitive effects in specific areas, such as the concept of variable. Because this body of research involves a great variety of variables, generalization is more difficult. These variables include school levels, students' ages, sex and ability levels, mathematical content, and computer language. The full impact of computer programming is still to be determined (Charp, 1981). Replication of several studies using new technologies and methodologies, and taking advantage of research findings to this point might be of value (Camp & Marchionini, 1981).

Uncontrolled variables and effects found in these studies are major concerns regarding the validity of the findings. Clark (1985) argues that the major uncontrolled effects are different instructional methods, content and/or novelty, and different or the same instructor teaching the control and experimental groups. The major threat is to the external validity of the results; that
is, how much we can generalize the findings. In other words, to what population, variables, and situations can the findings be generalized. It is important to look at the causes as well as the effects; we need to know exactly what makes the difference.

Another limitation of these studies is related to some of the findings regarding novice and expert programmers. Most investigators have failed to acknowledge the differences between novice and expert programmers and have made claims that are not in accordance with their short-term treatments. Most of these studies have examined high-level cognitive transfer or outcomes, expected to emerge only when the student has high-level programming skills. However, because of the short-term duration of the treatments, from six months to a year in most cases, the levels attained by the students in these studies are low (Pea & Kurland, 1984). This is a mismatch of "treatment and transfer assessments, because of the failure to identify the kind of transfer and their relationship with different levels of programming skills" (Pea and Kurland, 1984, p. 12).
Learning and Memory

The purpose of this section is to provide background information regarding the constructs of learning and memory. It is divided in four parts: defining learning, distinction between learning and performance, modern conception of learning, and memory and information storage.

Defining Learning

Bower and Hilgard (1981) point out that while it is extremely difficult to formulate a satisfactory definition of learning, this difficulty is not a source of controversy between theories which attempt to explain the learning process; however, the controversy is over fact and interpretation of learning. For example, behavioral theories of learning focus primarily on changes in behavior (changes in what the learner does), but the cognitive theories of learning focus on changes in content or structure of knowledge in memory (changes in what the learner knows). In general, learning is defined as "the relative permanent change in a person's knowledge or behavior due to experience" (Mitzel, Best and Robinowitz, 1982, p. 1042). As they state, this definition of learning has three components: "(1) the duration of change is long-term rather than short-term; (2) the
focus of change is the content and structure of knowledge in memory or the behavior of the learner; (3) the value of the change is the learner's experience in the environment rather than fatigue, motivation, drugs, physical conditions, or psychological intervention" (p. 1040). This definition indicates the relationship that exists between learning and memory.

**Distinction between Learning and Performance**

A distinction must be made between learning and performance, since person's performance may not always give an indication of what he/she has learned (Mitzel, Best and Robinowitz, 1982). "Learning" refers to the acquisition of knowledge of behavior, while "performance" refers to actual behavior that a learner exhibits on a given occasion. For example, performance on a posttest may be affected by where the test is administered, the stress experienced by the student, and/or the format of the test (Mitzel, Best and Robinowitz, 1982). Learning, however, can not be observed directly, and its presence must be inferred from observing changes in performance (Drewe, 1976).

**Modern Conception of Learning**

Most investigators have shifted from a "connectionist" view of learning to an "information-
processing" view of learning (Gagne, 1973). The
connectionist view states that repetition increases the
strength of learned connections (the more repetition, with
limits, the better the learning), and repetition is needed
to insure remembering.

Several investigators have challenged this view of
learning and memory. Gagne, Mayor, Garstens and Paradise
(1962) conducted a study in which seventh-grade students
learned about the addition of integers. One group of
students was given four or five times as many practice
problems on each of ten skills as were given to another
group. No difference was found in students' final
performance.

Gibson (1964) also studied the effect of practice on
students' performance. The sample included third- and
fourth-grade students who learned how to read numbers in
decimal form, write numbers in decimal form, and locate
decimal numbers on a number line. One group of students
received a total of ten practice examples for each skill
(intermediate amount of practice), a second group received
25 for each (large amount of practice), and a third group
received none at all (no practice). Different amount of
practice was found to have no effect on students' performance at the end of the treatment and five weeks
after treatment.

In these studies, practice was not found to be a
significant factor affecting students' retention. Gagne (1973) suggests that if practice is not a factor, then students' cognitive entry should account for their differences in retention of learning.

Modern theorists consider the connectionist view of learning as too simple (Gagne, 1973). The modern point of view tends to regard learning as a complex combination of processes taking place in the students' nervous system (Sywlester, 1985). This view is often called an "information-processing" conception of learning. According to this theory, information is first processed by the sensory register, then it enters the short-term memory, and finally it is transferred to the long-term memory. These three stages are discussed in the next section.

Memory and Information Storage

Memory plays an important role in learning; without memory, students would have no knowledge of past events (Keele, 1973). The study of learning and memory may be regarded as research into how past experiences are able to affect information processing and the resultant performance (Drewe, 1976). Drewe states that "studies of learning tend to concentrate on the acquisition of knowledge about events and those of memory on how retention or retrieval of information is affected with
time" (p. 67). Thus, studies in learning and studies in memory represent a difference in orientation rather than subject matter investigated (Drewe, 1976).

Mitzel, Best and Robinowitz (1982) indicate that memory consists of two processes: storing information and retrieving information. Psychologists have divided information storage into three stages: sensory register, short-term memory, and long-term memory.

*Sensory Register.* Information is first processed in the sensory memory. It is the act of taking information and absorbing it into the sensory register, two examples being sight, and sound (Keele, 1973). Information in this storage decays quickly (within a second) and would be irrevocably lost if not transferred to a "short-term memory store" (Neisser, 1967). Neisser adds that this process is considered very selective and only information that the person recognizes and considers important is passed on to other stages of memory. Also, Atkinson and Shiffrin (1968) indicate that this process is affected by our previous knowledge and attention.

*Short-term Memory.* The short-term memory is considered the center of activity in the information processing system (Keppel and Underwood, 1962; Reitman, 1971; Waugh and Norman, 1965). It is more selective and
slightly more permanent than sensory memory, and is demonstrated when the student recalls, for example, a telephone number, an address, directions for assembling a toy, or similar information just long enough for immediate use. Keele (1973) indicates that material in short-term memory will disappear in 10 to 20 seconds if it is not repeated or practiced.

According to Atkinson and Shiffrin (1968), an important process takes place in the short-term memory. It is an internal reviewing mechanism, called a rehearsal buffer, which organizes and rehearse the information. Information is not transferred in raw form, but it is transformed or coded in some way which will make it easier to remember (Gagne, 1973). If practice continues, the information may be transferred to long-term memory and permanently stored. Miller's (1956) generally accepted theory maintains that short-term memory can hold on to only seven items, plus or minus two items at a time. Information in the short-term memory can come from either the sensory memory or the long-term memory.

Long-term Memory. When the products of learning persist beyond an immediate occasion for their use, long-term memory is observed. It has a vast capacity for information. The information stored in long-term memory is highly organized and relatively permanent.
Keele (1973) states that this information "has been sifted, rehearsed, and coded and is now ready to find its place within a body of remembered experiences, which is, for the most part, constantly being revised and organized as new material flows into it". (p. 83). Also, information in the long-term memory depends on the action of the short-term memory for updating and reorganization (Ellis, Goggin and Parente, 1979).

Another process in the long-term memory is "retrieval," which comes into place when a person attempts to remember something (Gagne, 1973). It is what a person does when asked to recall some information learned in the past. The person uses the processes of searching and finding previously learned information in his/her memory.

**Summary**

There are three important stages related to learning and memory: sensory register, short-term memory, and long-term memory. The processes of coding and retrieval play an important role in this system. The way the information is reviewed, rehearsed, organized and transformed might affect the future use of the information. The use of computer programming to model mathematical concepts might help in facilitating the processes involved in this system. It might help students to search for information, examine mental images, and draw
inferences from learned information within their memory (Sywlester, 1985).

In general, the importance of the concept of variable in mathematics is supported by the literature reviewed in this chapter. Several studies indicate that the concept of variable is a troublesome concept for many students. Computer programming provides a concrete model of the concept of variable that might provide for students' learning of mathematics concepts and skills and for the retention of this learning. The current study was designed with respect to the background information presented in this chapter.
CHAPTER 3

Method

The purpose of this study was to investigate whether there was a significant difference in understanding the concept of variable and in attitudes toward mathematics among sixth grade students who used LOGO graphics (computer group), students who used a textbook-based approach (textbook group), and students who received no instruction to learn the concept of variable (control group). In addition, the strength of the relationship between students' posttest and retention test scores and their pre-treatment scores was analyzed.

Pre-treatment Design

The study was conducted in two phases, pre-treatment and treatment. The Test of Logical Thinking (TOLT), Robustness Semantic Differential (RSD), and Comprehensive Test of Basic Skills (CTBS) (described in the next section) were administered to all the students included in the sample prior to any work on the computers. Immediately following the administration of these instruments, all the students participated in the same pre-treatment activities.

Table 1 presents a summary of the pre-treatment design. The code used is as follows. The letter P refers to the process of measurement. P(1) refers to the TOLT,
P(2) refers to the RSD, and the P(3) refers to the CTBS. 
PRE-TRT represents exposure to the pre-treatment lessons. 
The left to right dimension indicates the temporal order 
over time of the different events.

Table 1
Summary of Pre-treatment Design

<table>
<thead>
<tr>
<th>P(1)</th>
<th>P(2)</th>
<th>P(3)</th>
<th>PRE-TRT</th>
</tr>
</thead>
</table>

**Research Design**

The design of the study was a posttest only, control 
group design with random assignment of subjects to 
treatment conditions. It included a control group and two 
experimental groups, each given a posttest and a retention 
test.

After the pre-treatment, the students in each 
mathematics section were assigned randomly to one of the 
experimental or computer groups. Random assignment at 
this point avoided possible bias from the teachers and 
investigator regarding students' activities prior to the 
treatment. Random assignment of students to sections was 
not possible because of scheduling problems; however, 
students in each mathematics section were randomly 
assigned to one of three treatments: computer group,
textbook group, and the control group. The textbook group and computer group received instruction on the concept of variable, and the control group received no instruction on the concept of variable.

Table 2 presents a summary of the experimental design of this study. The code used is as follows: $X(1)$ represents exposure to computer instruction, $X(2)$ represents exposure to textbook-based instruction, and $CL$ represents the control group with no instruction on the concept of variable. The effects of these experimental variables were measured using the Understanding of the Concept of Variable Instrument (UCVI). An $O$ refers to the process of measurement. $O(1)$ refers to the posttest, and the $O(2)$ refers to the retention test. The $R$ indicates the random assignment of students to the different groups. The left to right dimension indicates the temporal order over time of the different events.

Table 2

<table>
<thead>
<tr>
<th></th>
<th>$X(1)$</th>
<th>$O(1)$</th>
<th>$O(2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R$</td>
<td>$X(1)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$R$</td>
<td>$X(2)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$R$</td>
<td>$CL$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

Sixth grade students were selected as subjects because they have not yet been introduced formally to the concept of variable and have the prerequisite mathematical knowledge and logical thinking level required in the treatment activities. Most students showed comprehension of the following prerequisites: operations with whole numbers, recognizing different geometric shapes, estimating measures of angles, and measuring angles with a protractor. Four sixth grade mathematics sections, heterogeneously grouped, participated in the study. Table 3 presents a summary of treatment groups by school and section. Three sections taught by the same teacher were from Glen Oaks Middle School (61 students), East Baton Rouge Parish, and one section taught by another teacher was from the Louisiana State University (LSU) Lab School (28 students). A total of 89 students (female=47, male=42) participated in the study. The Glen Oaks Middle School students were of average and below average mathematics ability level based on their scores in the Comprehensive Test of Basic Skills (CTBS) and from a low socioeconomic level. About 80 percent of the students qualify to get free lunch at the school. The LSU Lab School students were of average and above average mathematics ability level based on their scores in the CTBS, and from a middle or high socioeconomic level.
Table 3

Description of Sample by Treatment

<table>
<thead>
<tr>
<th>School</th>
<th>Treatment</th>
<th>Computer</th>
<th>Textbook</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glen Oaks</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Section 1</td>
<td>21</td>
<td>6</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>Section 2</td>
<td>19</td>
<td>4</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>Section 3</td>
<td>21</td>
<td>9</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>LSU Lab School</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Section 4</td>
<td>28</td>
<td>9</td>
<td>8</td>
<td>11</td>
</tr>
<tr>
<td>Total</td>
<td>89</td>
<td>28</td>
<td>31</td>
<td>30</td>
</tr>
</tbody>
</table>
It was suspected that students were novice programmers. Studies of computer use in elementary schools suggest that children do not spend enough time doing computer programming to make them experts. This is in contrast with the long sessions and large amounts of time expert programmers spend in computer programming before they become expert programmers (Kurland, Mawby & Cahir, 1984). According to Kurland, Mawby and Cahir (1984) at least 500 hours of computer access are necessary to become an expert programmer.

**Instrumentation**

The study was conducted in two phases, pre-treatment and treatment. The instructional lessons and instruments used in each phase are described below. All pre-treatment and treatment lessons included the following items: lesson focus, materials, group size, time for activities, teaching notes (suggestions and instructions for the teacher to follow), and handout activities (seatwork or computer work) for the students.

**Pre-treatment Instructional Lessons**

In the pre-treatment phase, five fifty-minute lessons (see Appendix A) were used to teach the two experimental and control groups LOGO basic primitive commands without using variables. Activities for these lessons were
adapted from the following books: *Mindstorms* (Papert, 1980), *My students use computers* (Hunter, 1984), *LOGO discoveries* (Moore, 1984b), and *Geometry problems for LOGO discoveries* (Moore, 1984a). Students in the experimental and control groups learned LOGO graphics programming activities before they learned about the concept of variable in the treatment section.

The students followed specific experiences in order to learn LOGO basic commands. These lessons were taught by the investigator. The focus for each lesson was as follows:

**Lesson 1:** To introduce the turtle and the turtle graphics world, introduce LOGO primitive commands (FORWARD, BACK, RIGHT, LEFT, PENUp, PENDOWN, CLEARSCREEN, SHOWTURTLE, HIDETURTLE and HOME), use shortcuts to write these commands (FD, RT, BK, LT, PU, FD, ST, HT and CS), and make the turtle draw designs;

**Lesson 2:** To make the turtle draw designs, record the commands needed to draw a design, and sketch the designs the turtle draws;

**Lesson 3:** To enter a LOGO procedure into the computer, discover the result of executing the procedure, and edit commands;

**Lesson 4:** To modify a LOGO procedure to make a new but similar task, name the new procedure, and develop
a procedure using the LOGO primitive command REPEAT;

Lesson 5: To develop polygons using the REPEAT command and the Total Turtle Trip Theorem.

Treatment Instructional Lessons

The treatment consisted of two different approaches, computer programming and textbook-based instruction on the concept of variable. The following two sections describe the activities included in each approach.

Computer Programming Lessons. The computer group received instruction on the concept of variable through five fifty-minute computer programming lessons (see Appendix B). Activities for these lessons were adapted from the same computer programming books used for the pre-treatment instructional lessons. In these activities, students wrote, ran, and edited programs using one, two or three variable input to draw geometric figures. The emphasis was on teaching the concept of variable as it is used in mathematics through computer programming. The focus for each lesson was as follows:

Lesson 6A: To develop and debug (edit) LOGO procedures that involve one variable;

Lesson 7A: To develop and debug (edit) LOGO procedures that involve one or two variables;

Lesson 8A: To develop and debug (edit) LOGO
procedures that involve one or two variables;
Lesson 9A: To work with LOGO procedures involving
repetition and two or three variables;
Lesson 10A: To develop and debug (edit) LOGO
procedures that involve one or two variables,
repetition and increments.

All the computer programming lessons used in
pre-treatment and treatment phases of this study were
analyzed and evaluated by two middle school computer
programming teachers in terms of readability, difficulty,
and time needed to perform the activities. These teachers
provided insightful comments with regard to topic
suggestions, reading level, and programming suggestions.
Adaptations were made according to their suggestions.
Also, these lessons were field tested with eight sixth
grade students of varying mathematical ability and
backgrounds. Changes were made in terms of readability,
length of the activities, and difficulty level.

Textbook-based Lessons. The textbook group
received instruction on the concept of variable through
another series of lessons without computer activities.
Another series of five fifty-minute lessons (see Appendix
C) was developed to teach the concept of variable using a
textbook-based approach. These lessons were adapted from
Addison-Wesley_Mathematics: Teachers' Edition, Book 6 and
7 (Eicholz et al., 1987). Emphasis was placed on helping students understand the concept of variable as it is used in mathematics. The focus for each lesson was as follows:

Lesson 6B: To introduce the concept of variable and solve equations by finding a number replacement for the variable that will make the equation true;

Lesson 7B: To introduce the concept of variable as it is used in expressions and evaluate expressions for given replacements of the variable;

Lesson 8B: To evaluate expressions for given replacements of the variable;

Lesson 9B: To provide practice regarding the concept of variable and evaluate expressions for given replacements of the variable;

Lesson 10B: To provide practice regarding the concept of variable, and evaluate expressions for given replacements of one and two variables.

The five lessons designed to teach the concept of variable without using computer programming instruction were analyzed and evaluated by two sixth grade mathematics teachers. These lessons were evaluated in terms of readability, difficulty level, and time needed by sixth graders to perform the different tasks. Changes were made according to the teachers' suggestions.
Pre-treatment Instruments

Three instruments were used to gather the data necessary from each student: Comprehensive Test of Basic Skills (CTBS), Test of Logical Thinking (TOLT), and Robustness Semantic Differential (RSD). A description of each of these three instruments follows.

Comprehensive Test of Basic Skills. As a measure of mathematics achievement, the CTBS (Mathematics Section, Form V) was administered to all students. The CTBS is a standardized achievement test which assesses students' mathematics achievement in computation, concepts, and applications, and is known to predict mathematics achievement accurately and reliably. Mathematics achievement is defined as the sum of the points earned on this instrument. The content of the test was examined carefully and compared with other mathematics achievement tests to determine whether or not it was appropriate for the purpose of the study. The test was determined as valid for the objective of the study.

The developers state that several procedures were used to assure that the basic skills selected for measurement were appropriate for the designated grade level and for students throughout the nation (California Test Bureau, 1982). The test items were written by classroom teachers of the grade for which the test was
designated in cooperation with curriculum and testing specialists. The test was edited by trained item writers using established rules for the writing of items. The split-half reliability estimate (Kuder-Richardson Formula 20) used for testing the internal consistency of the test was .86 (California Test Bureau, 1982).

Test of Logical Thinking. The cognitive development of students is one of the factors that is considered to account for variation among students' learning. The TOLT (form A), developed by Tobin and Capie (1980) and used to assess students' cognitive level, is a paper and pencil test in which students are asked to solve word problems and justify their results (see Appendix D). It was selected because of its validity, reliability and easy administration in a group setting. Coefficient Alpha for TOLT based on the total sample of 682 students was .85 (Tobin & Capie, 1980). Item difficulty ranged from .18 to .41 with an average of .35. Five modes of reasoning were included: proportional reasoning (items 1 and 2), controlling variables (items 3 and 4), probabilistic reasoning (items 5 and 6), correlation reasoning (items 7 and 8), and combinatorial reasoning (items 9 and 10).
Robustness Semantic Differential. The students' affective level is one of the factors considered to account for variation among students' learning (Bloom, 1976). The RSD, which is a valid, reliable, and easy to administer scale, was used to assess students' attitudes toward mathematics. The target objects included in the scale were selected according to the instructional elements involved in the study. Students were asked to respond to this scale which was structured as a semantic differential and included six subscales or target objects: RSD1 (mathematics teacher), RSD2 (computer), RSD3 (mathematics activities), RSD4 (mathematics learning activities with the computer), RSD5 (interaction with the mathematics teacher), and RSD6 (interaction with the computer). A five-point scale with 10 pairs of bipolar adjectives assessing evaluative, activity, and potency dimensions of each target object were used (see Appendix E). Student scores can range from 10 to 50 for a particular subscale or target object.

Licata and Willower (1975, 1978) developed the construct of environmental robustness as a school climate variable based on the work of Goffman (1959, 1961, 1963, 1967). The idea of environmental robustness is relies on the notion that social situations can be understood using theatrical analogies which identify actors, plot, setting and audience (Licata and Wildes, 1980). It is understood
as a theoretical construct in terms of these theatrical analogies and defined as the perceived "dramatic" content of the school structure for particular audiences such as teachers, students, parents or administrators (Willower and Licata, 1975). Using a factor analysis of the term "drama," Licata and Willower (1978) identified ten adjective pairs which contribute to the meaning of the term "drama." The adjective pairs were used in order to develop an operational definition of the perception of drama or environmental robustness in schools: interesting-boring, challenging-dull, active-passive, unusual-usual, powerful-weak, thrilling-quieting, important-unimportant, fresh-stale, meaningful-meaningless, and action-packed-uneventful (Licata and Willower, 1978). Adjectives which connote robustness are interesting, challenging, active, unusual, powerful, thrilling, important, fresh, meaningful, and action-packed. Adjectives which connote relative lack of robustness are boring, dull, passive, usual, weak, quieting, unimportant, stale, meaningless, and uneventful.

These adjectives pairs were operationalized with seven point semantic differential scales (Osgood, et al., 1957). The student's response to a particular school concept is measured in terms of rating of these 10 adjectives. The higher the score, the more robust the target object. The test-retest coefficients for the total
instrument were .77 (Pearson) and .78 (Spearman) (Licata and Willower, 1978). Predictive validity was demonstrated for each of the ten items based on their ability to discriminate significantly between the terms "dramatic" and "not dramatic" (Licata and Willower, 1978).

Using a sample of 592 elementary, middle and high school teachers, Morris (1986) found Cronbach alpha reliability coefficients ranging from .84 to .92 for a six RSD target objects. The investigator found Pearson r correlation coefficients ranging from .44 to .77 between middle-school teachers’ perceptions of student robustness and student achievement.

Post-treatment Instrument

The Understanding of the Concept of Variable Instrument (UCVI) developed by the investigator was used as a posttest and retention test (see Appendix F). This 32-item instrument was based on a protocol developed by Collis (1975) to assess students’ understanding of the concept of variable from which items 15, 23, 25, 27 and 29 were taken. The UCVI requires that students think about variables as representing a range of values, in addition to describing the degree to which changes in one set relates to changes in another. Also, this test requires students to evaluate expressions by substituting values for the variables. The format of this test requires
students to decide whether mathematical statements are true always, sometimes or never, and explain, if their answer is "sometimes," when the mathematical statement is true. The instrument was dichotomously scored. When the answer is sometimes, the student should also provide the correct explanation to receive one point.

Collis found that students ranging in age from 10-15 years have no difficulty in understanding the meaning of the questions or symbols presented in the test, and that they accept that a letter may be used to stand for a number. Bell et al. (1983) state, "It is not necessary to have been taught algebra in a formal sense to be able to interpret questions such as the one included in this test" (p. 139).

Items 13, 14, 17, 18, 19, 20, 25, 26, 29 and 30 require students to assume that C and D though being different letters, can represent the same number. These items demand the use of the idea of both C and D running through the whole set of possible numbers and also the willingness to tolerate the ambiguous answer "sometimes" (Bell et al., 1983). Items 9, 10, 23 and 24 are considered of greater difficulty. In these items the problem is remembering zero as a possible value for C. Zero is regarded as a special number and often forgotten in such cases (Bell et al., 1983).

The Understanding of the Concept of Variable
Instrument was field tested and revised. The field test group included 26 sixth grade students of average and above average mathematical ability (based on their California Achievement Test scores). Also, the test was revised and evaluated by two sixth grade mathematics teachers. An item analysis using the item reliability index provided the following statistics. The mean of the test scores was 19.1 and the standard deviation was 5.19. The item difficulty level ranged from .32 to 1.00. The item difficulty mean of the test scores based on all items was .63, the standard deviation was .68, and the reliability index (Alpha) was .83. Most item discrimination indices ranged from .5 to 1.0. The internal consistency of the test was found to be high. Split-half reliability estimate was used for this purpose. Odd and even numbers were correlated. The internal consistency index was (Alpha) .93 and the reliability of the entire test was .96.

Additional evidence for the concurrent validity of the test was provided by correlating the teacher’s predicted scores for each student with the students’ actual scores in the test. The correlation coefficient for the predicted scores and the actual scores was .73.
Procedures

This section presents the procedures used in pre-treatment and treatment phases of the study. Pertinent details and further information can be found in the following discussion.

Pre-treatment

After these instruments were administered, all students in the sample were taught the pre-treatment lessons. For this purpose they worked through five fifty-minute lessons (one per day). These lessons were taught by the investigator. At the end of the pre-treatment, the students in each mathematics section were assigned randomly to one of the experimental or the control groups.

Treatment

Immediately following pre-treatment and random assignment of students to groups, treatment started, consisting of five fifty-minute lessons on five days involving the concept of variable. Each student worked as independently as possible in order to control for teacher effect on students' performance. The activities provided clear and detailed instructions to allow for students' independent work.
The students in the computer group were taught the concept of variable through five fifty-minute lessons using LOGO graphics computer programming. Students worked independently using the computers but were allowed to share ideas with each other. In this case the investigator tried to make certain each student understood any idea he/she borrowed from another student. The investigator supervised students' learning through different programming activities.

The students in the textbook group were taught the concept of variable through five fifty-minute lessons involving a textbook-based approach and no computer activities. These lessons were taught by the regular classroom teacher. The students worked as independently as possible through paper-and-pencil activities. Teacher instruction was kept to a minimum. The control group received no instruction on the concept of variable during the experiment but were involved in other mathematics-related activities assigned by the regular classroom teacher during the five fifty-minute treatment time periods.

At the end of treatment, the UCVI and the RSD were administered to all the students in order to assess their understanding of the concept of variable, and attitudes toward mathematics and computer programming. After three weeks the UCVI was readministered to assess students'
retention of learning.
CHAPTER 4

Results

The purpose of this study was to investigate whether there was a significant difference in understanding the concept of variable and in attitudes toward mathematics among sixth grade students who used a computer programming approach, students who used a textbook-based approach, and students who received no instruction. In addition, the strength of the relationship between students' posttest and retention test scores and their pre-treatment was also studied.

Pre-treatment scores were analyzed prior to assessing posttest treatment effects. The results of an analysis of variance indicated no significant differences between treatment groups (computer, textbook and control) on the pre-treatment measures of mathematics achievement, logical thinking, and attitudes toward mathematics (see Appendix G). The treatment groups appeared to be equivalent in terms of these factors at the beginning of the experiment. Treatment group means and standard deviations for each pre-treatment measure (CTBS, TOLT and RSD concepts) are given in Table 4.

The results of the tests of hypotheses are presented in five sections. The first section presents the results of the tests of hypotheses related to treatment effects on students' understanding of the concept of variable at the
Table 4
Means and Standard Deviations for Pre-treatment Measures by Treatment Group

<table>
<thead>
<tr>
<th>Instrument</th>
<th>Treatment n</th>
<th>TOLT</th>
<th>CTBS</th>
<th>RSD1</th>
<th>RSD2</th>
<th>RSD3</th>
<th>RSD4</th>
<th>RSD5</th>
<th>RSD6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer</td>
<td>28</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>0.7</td>
<td>22.7</td>
<td>27.8</td>
<td>43.6</td>
<td>29.2</td>
<td>40.9</td>
<td>29.1</td>
<td>42.8</td>
<td></td>
</tr>
<tr>
<td>SD</td>
<td>1.3</td>
<td>8.5</td>
<td>7.6</td>
<td>5.6</td>
<td>8.0</td>
<td>7.8</td>
<td>11.1</td>
<td>6.0</td>
<td></td>
</tr>
<tr>
<td>Textbook</td>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>0.8</td>
<td>21.8</td>
<td>26.7</td>
<td>43.6</td>
<td>28.3</td>
<td>40.9</td>
<td>28.5</td>
<td>41.8</td>
<td></td>
</tr>
<tr>
<td>SD</td>
<td>1.2</td>
<td>7.9</td>
<td>10.4</td>
<td>6.7</td>
<td>11.7</td>
<td>9.4</td>
<td>11.2</td>
<td>9.3</td>
<td></td>
</tr>
<tr>
<td>Control</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>0.9</td>
<td>24.2</td>
<td>29.3</td>
<td>41.0</td>
<td>29.3</td>
<td>40.2</td>
<td>31.4</td>
<td>42.4</td>
<td></td>
</tr>
<tr>
<td>SD</td>
<td>1.0</td>
<td>9.5</td>
<td>7.1</td>
<td>6.5</td>
<td>7.1</td>
<td>8.6</td>
<td>7.5</td>
<td>6.9</td>
<td></td>
</tr>
</tbody>
</table>

Note. TOLT = Test of Logical Thinking; CTBS = Comprehensive Test of Basic Skills; RSD1 = mathematics; RSD2 = computer; RSD3 = mathematics activities; RSD4 = mathematics activities with the computer; RSD5 = interaction with the teacher; RSD6 = interaction with the computer.
end of the study. The second section presents the results of the tests of the hypotheses related to treatment effects on students' understanding of the concept of variable three weeks after treatment. The third section presents the results of the test of the hypothesis related to treatment effects on students' attitudes toward mathematics at the end of the treatment. In the fourth section, results related to the relationship between students' posttest scores and their pre-treatment scores are presented. The final section presents results related to the supplemental analyses. The data were analyzed using Statistical Analysis System (SAS Institute, Inc., 1985) procedures. The results provided by the SAS GLM and DUNCAN'S procedures which were used for testing each hypothesis are presented in Appendix G.

**Treatment Effects on Students' Understanding of the Concept of Variable**

Means and standard deviations for students' posttest scores by treatment group are shown in Table 5. This section presents the results of the tests of hypotheses related to treatment effects on students' understanding of the concept of variable at the end of the treatment. Before the hypotheses were tested, an analysis of variance was computed to assess treatment effect. The independent
### Table 5

**Mean Post-treatment Scores by Treatment Group**

<table>
<thead>
<tr>
<th>Treatment</th>
<th>n</th>
<th>Posttest</th>
<th>Retention test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer</td>
<td>28</td>
<td>9.46</td>
<td>11.46</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.32</td>
<td>5.79</td>
</tr>
<tr>
<td>Textbook</td>
<td>31</td>
<td>7.55</td>
<td>6.77</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.30</td>
<td>4.50</td>
</tr>
<tr>
<td>Control</td>
<td>30</td>
<td>6.00</td>
<td>6.57</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.69</td>
<td>4.67</td>
</tr>
</tbody>
</table>

**Note.** Understanding of the Concept of Variable Instrument was administered as posttest and retention test (three weeks after treatment).
variable was treatment, and the dependent variable was the post-treatment score on the UCVI. There was a significant main effect for treatment, $F(2, 86) = 4.39, p < .0151$.

**Hypothesis 1:** Students receiving programming instruction will not demonstrate greater understanding of the concept of variable on the posttest than students receiving textbook-based instruction. This hypothesis was tested by an orthogonal contrast, computer group versus textbook group. Duncan's multiple-range test was employed with the designated orthogonal contrast, computer group versus textbook group included in the model. The dependent variable was the post-treatment score on the UCVI at the end of treatment. The difference found was nonsignificant with $p > .10$. Although the mean score on the UCVI for the students receiving computer programming instruction was greater than the mean score for the students receiving textbook-based instruction, the null hypothesis was not rejected and the predictive hypothesis was not confirmed.
Hypothesis 2: Students receiving programming instruction will not demonstrate greater understanding of the concept of variable on the posttest than students receiving no instruction on the concept of variable (control group students).

This hypothesis was tested by an orthogonal comparison, computer group versus control group. Duncan's multiple-range test was employed with the designated orthogonal contrast, computer group versus control group included in the model. The dependent variable was post-treatment score on the UCVI at the end of treatment. There was a significant main effect for the contrast with \( p < .01 \). The hypothesis was rejected and the predictive hypothesis was confirmed. The mean score on the UCVI was significantly higher for students receiving computer programming instruction than the mean score for students receiving no instruction on the concept of variable.

Hypothesis 3: Students receiving textbook-based instruction will not demonstrate greater understanding of the concept of variable on the posttest than students receiving no instruction on the concept of variable (control group students).

This hypothesis was tested by an orthogonal contrast, textbook group versus control group. Duncan's multiple-range test was employed with the designated
orthogonal contrast, textbook group versus control group included in the model. The independent variable was post-treatment score in the UCVI at the end of treatment. The contrast was not significant, \( p > .10 \). The null hypothesis was not rejected and the predictive hypothesis was not confirmed.

**Treatment Effects on Students' Understanding of the Concept of Variable Three Weeks After Treatment**

Means and standard deviations for students' retention test scores by treatment group are shown in Table 5 (see page 116). This section presents the results of tests of hypotheses related to treatment effects on students' understanding of the concept of variable three weeks after treatment. Before the hypotheses were tested, an analysis of variance was computed to assess treatment effect. The independent variable was treatment, and the dependent variable was post-treatment score in the UCVI three weeks after treatment. There was a significant main effect for treatment, \( F (2, 86) = 10.28, p < .0001 \).
Hypothesis 4: Students receiving programming instruction will not demonstrate greater understanding of the concept of variable on the retention test (three weeks after treatment) than students receiving textbook-based instruction.

This hypothesis was tested by an orthogonal contrast, computer group versus textbook group. Duncan's multiple-range test was employed with the designated orthogonal comparison, computer group versus textbook group included in the model. The dependent variable was post-treatment score on the UCVI three weeks after treatment. There was a significant main effect for the contrast with \( p < .01 \). The hypothesis was rejected and the predictive hypothesis confirmed. The mean score in the UCVI for students receiving computer programming instruction was significantly higher than the mean score for students receiving textbook-based instruction.

Hypothesis 5: Students receiving programming instruction will not demonstrate greater understanding of the concept of variable on the retention test (three weeks after treatment) than students receiving no instruction on the concept of variable (control group students).

This hypothesis was tested by an orthogonal contrast, computer group versus control group. Duncan's multiple-range test was employed with the designated
orthogonal contrast, computer group versus control group included in the model. The dependent variable was post-treatment score on the UCVI three weeks after treatment. There was a significant main effect for the contrast with \( p < .01 \). The hypothesis was rejected and the predictive hypothesis was confirmed. The mean score on the UCVI for the group receiving programming instruction was significantly higher than the mean score for the group receiving no instruction on the concept of variable. This difference seems to be educationally as well as statistically significant.

**Hypothesis 4:** Students receiving textbook-based instruction will not demonstrate greater understanding of the concept of variable on the retention test (three weeks after treatment) than students receiving no instruction on the concept of variable (control group students).

This hypothesis was tested by an orthogonal contrast, textbook group versus control group. Duncan's multiple-range test was employed with the designated orthogonal contrast, textbook group versus control group designated in the model. The dependent variable was post-treatment score on the UCVI three weeks after treatment. There was a nonsignificant main effect for the contrast with \( p > .10 \). The null hypothesis was not rejected and the predictive hypothesis was not confirmed.
Treatment Effect on Students' Attitudes Toward Mathematics

The RSD, which included six concepts, was administered again after the treatment. Means and standard deviations for students' scores for each of the RSD concepts are shown in Table 6. This section presents results of the test of the hypothesis related to treatment effects on students' attitudes toward mathematics at the end of the treatment.

Hypothesis 7: Students receiving programming instruction will have a more positive attitude toward mathematics at the end of the treatment than students receiving textbook-based instruction or no instruction on the concept of variable.

In order to compare the effect of using a computer programming approach, a textbook-based approach, and a non instruction approach to students' attitudes toward mathematics at the end of the treatment, an analysis of variance was performed, with the independent variable being treatment. The dependent variables were post-treatment scores for each RSD concept. There were no significant differences among students' attitudes toward any of the six target objects included in the RSD (mathematics teacher, computer, mathematics activities, mathematics activities with the computer, interaction
Table 6

**Mean RSD Concept Scores by Treatment Group**

<table>
<thead>
<tr>
<th>RSD</th>
<th>Treatment</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Computer</td>
<td>28</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M</td>
<td></td>
<td>26.0</td>
<td>39.7</td>
<td>28.7</td>
<td>37.4</td>
<td>30.7</td>
<td>38.3</td>
</tr>
<tr>
<td>SD</td>
<td></td>
<td>10.0</td>
<td>8.6</td>
<td>8.1</td>
<td>7.0</td>
<td>7.1</td>
<td>8.1</td>
</tr>
<tr>
<td>Textbook</td>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M</td>
<td></td>
<td>25.4</td>
<td>43.5</td>
<td>27.2</td>
<td>41.2</td>
<td>28.5</td>
<td>43.0</td>
</tr>
<tr>
<td>SD</td>
<td></td>
<td>11.1</td>
<td>8.4</td>
<td>12.4</td>
<td>11.3</td>
<td>11.9</td>
<td>7.8</td>
</tr>
<tr>
<td>Control</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M</td>
<td></td>
<td>26.3</td>
<td>41.9</td>
<td>27.3</td>
<td>40.1</td>
<td>28.0</td>
<td>41.6</td>
</tr>
<tr>
<td>SD</td>
<td></td>
<td>9.2</td>
<td>10.3</td>
<td>8.7</td>
<td>11.2</td>
<td>8.1</td>
<td>10.1</td>
</tr>
</tbody>
</table>

**Note.** The scores included in this table are from the RSD given at the end of treatment.

Maximum score for each RSD subscale = 50.

RSD = Robustness Semantic Differential; RSD1 = mathematics teacher; RSD2 = computer; RSD3 = mathematics activities; RSD4 = mathematics activities with the computer; RSD5 = interaction with the teacher; RSD6 = interaction with the computer.

The higher the score, the greater the robustness of the six target objects or concepts included in the RSD.
with the mathematics teacher, and interaction with the computer. The null hypothesis was not rejected and the predictive hypothesis was not confirmed. Results of the analysis of variance for each RSD target object are presented in Tables 7, 8, 9, 10, 11, and 12.

Supplemental Analyses

Supplemental analyses appropriate to the data were completed as suggested by the initial data analysis results. These analyses include examining the strength of the relationship between students' pre-treatment scores and their post-treatment scores, intercorrelations between students' pre-treatment scores, RSD subscale scores, multiple regression of students' post-treatment scores on their pre-treatment scores, reliability analyses of RSD and UCVI scores, and a suggested model to increase R-Square.

Relationship Between Students' Pre-treatment and Posttest Scores

The strength of the relationship between students' understanding of the concept of variable at the end of the treatment and their logical thinking, mathematics achievement, and attitudes toward mathematics was inhered from the significance of the correlation between students' posttest scores and their scores in each of the
Table 7

**Analysis of Variance for RSD1 Scores**

Dependent variable: Pre-treatment scores for RSD1

<table>
<thead>
<tr>
<th>Source of Variance</th>
<th>df</th>
<th>Sum of Squares</th>
<th>Mean Square</th>
<th>F</th>
<th>p</th>
<th>R²</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>2</td>
<td>29.66</td>
<td>14.83</td>
<td>.19</td>
<td>.83</td>
<td>.004</td>
</tr>
<tr>
<td>Error</td>
<td>86</td>
<td>6724.45</td>
<td>78.19</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Corrected Total</td>
<td>88</td>
<td>6754.11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note.** RSD1 = Robustness Semantic Differential with target object being the mathematics teacher.
### Table 8

**Analysis of Variance for RSD2 Scores**

---

**Dependent variable:** Pre-treatment scores for RSD2

<table>
<thead>
<tr>
<th>Source of Variance</th>
<th>df</th>
<th>Sum of Squares</th>
<th>Mean Square</th>
<th>( F )</th>
<th>( p )</th>
<th>( R^2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>2</td>
<td>140.78</td>
<td>70.39</td>
<td>1.80</td>
<td>.17</td>
<td>.04</td>
</tr>
<tr>
<td>Error</td>
<td>86</td>
<td>3368.75</td>
<td>39.17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Corrected Total</td>
<td>88</td>
<td>3509.53</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>88</td>
<td>3509.53</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note.** RSD2 = Robustness Semantic Differential subscale with target object being the computer.
Table 9

Analysis of Variance for RSD3 Scores

<table>
<thead>
<tr>
<th>Source of Variance</th>
<th>df</th>
<th>Sum of Squares</th>
<th>Mean Square</th>
<th>F</th>
<th>p</th>
<th>R²</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>2</td>
<td>18.95</td>
<td>9.47</td>
<td>.11</td>
<td>.89</td>
<td>.0003</td>
</tr>
<tr>
<td>Error</td>
<td>86</td>
<td>7277.50</td>
<td>84.62</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Corrected Total</td>
<td>88</td>
<td>7296.45</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note. RSD3 = Robustness Semantic Differential subscale with target object being mathematics activities.
Table 10

**Analysis of Variance for RSD4 Scores**

<table>
<thead>
<tr>
<th>Source of Variance</th>
<th>df</th>
<th>Sum of Squares</th>
<th>Mean Square</th>
<th>F</th>
<th>p</th>
<th>R²</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>2</td>
<td>9.71</td>
<td>4.86</td>
<td>0.06</td>
<td>0.94</td>
<td>0.002</td>
</tr>
<tr>
<td>Error</td>
<td>86</td>
<td>6431.09</td>
<td>74.78</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Corrected Total</td>
<td>88</td>
<td>6440.80</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Note.* RSD4 = Robustness Semantic Differential subscale with target object being mathematics activities with the computer.
## Table 11

**Analysis of Variance for RSDS5 Scores**

<table>
<thead>
<tr>
<th>Source of Variance</th>
<th>df</th>
<th>Sum of Squares</th>
<th>Mean Square</th>
<th>F</th>
<th>p</th>
<th>R²</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>2</td>
<td>141.18</td>
<td>70.59</td>
<td>.69</td>
<td>.50</td>
<td>.02</td>
</tr>
<tr>
<td>Error</td>
<td>86</td>
<td>7277.50</td>
<td>84.62</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Corrected Total</td>
<td>88</td>
<td>7296.45</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note.** RSDS5 = Robustness Semantic Differential subscale with target object being interaction with the mathematics teacher.
Table 12

Analysis of Variance for RSD6 Scores

Dependent variable: Pre-treatment scores for RSD6

<table>
<thead>
<tr>
<th>Source of Variance</th>
<th>df</th>
<th>Sum of Squares</th>
<th>Mean Square</th>
<th>F</th>
<th>p</th>
<th>R²</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>2</td>
<td>15.58</td>
<td>7.79</td>
<td>.14</td>
<td>.87</td>
<td>.003</td>
</tr>
<tr>
<td>Error</td>
<td>86</td>
<td>4942.31</td>
<td>57.47</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Corrected Total</td>
<td>88</td>
<td>4957.89</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note. RSD6 = Robustness Semantic Differential subscale with target object being interaction with the computer.
pre-treatment measures (TOLT, CTBS, and RSD) by treatment group. Table 13 presents the correlations between students' posttest scores and each pre-treatment measure by treatment group. The following supplemental research questions were analyzed based on these correlations:

What is the relationship between students' posttest scores by treatment group and their scores on the affective scale?

The correlation between students' posttest scores and their scores for each RSD concept was tested by treatment group. The Pearson $r$ coefficient for the correlation between computer, textbook, and control group students' scores on the posttest and their scores on each RSD concept scores ranged from zero ($r = .02$) to low ($r = .25$) in magnitude and were positive in direction. The correlations between these two variables were not significant for students receiving computer programming instruction, students receiving textbook-based instruction and students receiving no instruction on the concept of variable.
Table 13

**Pearson r Correlation Between Pre-treatment and Posttest Scores by Treatment Group**

<table>
<thead>
<tr>
<th>Pretest</th>
<th>UCVI Posttest</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Computer</td>
</tr>
<tr>
<td></td>
<td>r</td>
</tr>
<tr>
<td>RSD1</td>
<td>.14</td>
</tr>
<tr>
<td>RSD2</td>
<td>.04</td>
</tr>
<tr>
<td>RSD3</td>
<td>.21</td>
</tr>
<tr>
<td>RSD4</td>
<td>.03</td>
</tr>
<tr>
<td>RSD5</td>
<td>.24</td>
</tr>
<tr>
<td>RSD6</td>
<td>.20</td>
</tr>
<tr>
<td>CTBS</td>
<td>.56</td>
</tr>
<tr>
<td>TOLT</td>
<td>.60</td>
</tr>
</tbody>
</table>

* p < .05. ** p < .01. *** p < .001.

**Note.** UCVI = Understanding of the Concept of Variable Instrument; RSD = Robustness Semantic Differential; RSD1 = mathematics; RSD2 = computer; RSD3 = mathematics activities; RSD4 = mathematics activities with the computer; RSD5 = interaction with the teacher; RSD6 = interaction with the computer; CTBS = Comprehensive Test of Basic Skills; TOLT = Test of Logical Thinking.
What is the relationship between students' posttest scores by treatment group and their scores on the mathematics achievement test?

The correlation between students' posttest scores and their scores on the mathematics achievement test was tested by treatment group. The Pearson $r$ coefficient for the correlation between students' scores on the posttest and their scores on the mathematics achievement test was significant for the students receiving computer programming instruction, $r = .56$, $p < .0002$. The Pearson $r$ coefficient for the correlation between the textbook group students' scores on the posttest and their scores in the mathematics achievement test was also significant, $r = .38$, $p < .03$. Finally, the Pearson $r$ coefficient for the correlation between the control group students' posttest scores and their mathematics achievement scores was also significant, $r = .71$, $p < .0001$.

What is the relationship between students' posttest scores by treatment group and their scores on the logical thinking test?

The correlation between students' posttest scores and their logical thinking scores was tested by treatment group. The Pearson $r$ coefficient for the computer group scores on the posttest and their scores on the logical thinking test was significant, with $r = .69$ and $p < .0007$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
The Pearson $r$ coefficient for the correlation between the textbook group students' posttest scores and their scores on the logical thinking test was significant, $r = .52, p < .003$. The Pearson $r$ coefficient for the correlation between control group students' scores on the posttest and their scores on the logical thinking test was not significant.

**Relationship Between Students' Pre-treatment and Retention Test Scores**

The relationship between students' understanding of the concept of variable three weeks after treatment and their logical thinking, mathematics achievement, and attitudes toward mathematics was inferred from the significance of the correlation between students' scores on the retention test and their scores on each of the pre-treatment measures by treatment group. Table 14 presents the correlations between students' retention test scores and their pre-treatment scores by treatment group. The following research questions were analyzed based on these correlations.

**What is the relationship between students' retention test scores by treatment group and their scores on the affective scale?**

The correlation between students' retention test scores...
Table 14

Pearson correlation between pre-treatment and retention test scores by treatment group

<table>
<thead>
<tr>
<th></th>
<th>UCVI Retention Test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Computer</td>
</tr>
<tr>
<td></td>
<td>L</td>
</tr>
<tr>
<td>RSD1</td>
<td>.24</td>
</tr>
<tr>
<td>RSD2</td>
<td>.19</td>
</tr>
<tr>
<td>RSD3</td>
<td>.14</td>
</tr>
<tr>
<td>RSD4</td>
<td>.12</td>
</tr>
<tr>
<td>RSD5</td>
<td>.12</td>
</tr>
<tr>
<td>RSD6</td>
<td>.03</td>
</tr>
<tr>
<td>CTBS</td>
<td>.56</td>
</tr>
<tr>
<td>TOLT</td>
<td>.53</td>
</tr>
</tbody>
</table>

* p < .05.  ** p < .01.  *** p < .001.

Note. UCVI = Understanding of the Concept of Variable Instrument; RSD = Robustness Semantic Differential; RSD1 = mathematics; RSD2 = computer; RSD3 = mathematics activities; RSD4 = mathematics activities with the computer; RSD5 = interaction with the teacher; RSD6 = interaction with the computer; CTBS = Comprehensive Test of Basic Skills; TOLT = Test of Logical Thinking.
scores and their scores on each of the RSD concepts was tested by treatment group. The Pearson $r$ coefficient for the correlation between computer, textbook, and control group students' retention test scores and their scores on each RSD concept ranged from zero ($r = .01$) to low ($r = .24$) in magnitude and were positive in direction. These correlations were not significant for students receiving computer programming instruction, students receiving textbook-based instruction or students receiving no instruction on the concept of variable.

What is the relationship between students' retention test scores by treatment group and their scores on the mathematics achievement test?

The correlation between students' retention test scores and their scores on the mathematics test was tested by treatment group. The Pearson $r$ coefficient for the correlation between retention test scores and mathematics achievement test scores was significant for students receiving computer programming instruction, $r = .56$, $p < .002$. The Pearson $r$ coefficient for the correlation between textbook group students' retention test scores and their mathematics achievement test scores was also significant, $r = .47$, $p < .007$. This was also the case for the control group students' scores in these two tests, $r = .69$, $p < .0001$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
What is the relationship between students’ retention test scores by treatment group and their scores on the logical thinking test?

The correlation between students’ retention test scores and their scores on the logical thinking test was tested by treatment group. The Pearson $r$ coefficient for the correlation between computer programming group students’ retention test scores and their logical thinking test scores was significant, $r = .53$, $p < .003$. This was also the case for the textbook and control group students’ scores in these two tests, with $r = .45$ and $p < .01$.

**Intercorrelations Between Pre-treatment Measures**

These supplemental analyses were based on the Pearson $r$ coefficients for the intercorrelations between the following pre-treatment measures: TOLT, CTBS, and RSD concepts. Table 15 presents the correlations between the pre-treatment measures for all students ($n = 89$). The correlations ranged from zero ($r = .01$) to moderately strong ($r = .69$) in magnitude and were positive in direction.

The intercorrelations between the RSD concept scores ranged from low ($r = .20$) to moderately strong ($r = .70$) in magnitude and were positive in direction. The only statistically significant correlation
Table 15

Pearson r Intercorrelation Between Pre-treatment Scores

<table>
<thead>
<tr>
<th>Instrument</th>
<th>TOLT</th>
<th>RSD2</th>
<th>RSD3</th>
<th>RSD4</th>
<th>RSD5</th>
<th>RSD6</th>
<th>RSD6</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTBS</td>
<td>.30</td>
<td>.08</td>
<td>.18</td>
<td>.01</td>
<td>.10</td>
<td>.05</td>
<td>.15</td>
</tr>
<tr>
<td>TOLT</td>
<td>---</td>
<td>.05</td>
<td>.08</td>
<td>.01</td>
<td>.17</td>
<td>.25</td>
<td>.22</td>
</tr>
<tr>
<td>RSD1</td>
<td>---</td>
<td>.25</td>
<td>.63</td>
<td>.27</td>
<td>.40</td>
<td>.20</td>
<td></td>
</tr>
<tr>
<td>RSD2</td>
<td>---</td>
<td>.12</td>
<td>.70</td>
<td>.21</td>
<td>.47</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RSD3</td>
<td>---</td>
<td>.27</td>
<td>.48</td>
<td>.27</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RSD4</td>
<td>---</td>
<td>.33</td>
<td>.64</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RSD5</td>
<td>---</td>
<td>.31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RSD6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(n = 89)

*p < .05.  ** p < .01.  *** p < .001.

Note. RSD = Robustness Semantic Differential; RSD1 = mathematics; RSD2 = computer; RSD3 = mathematics activities; RSD4 = mathematics activities with the computer; RSD5 = interaction with the teacher; RSD6 = interaction with the computer; CTBS = Comprehensive Test of Basic Skills; TOLT = Test of Logical Thinking.
coefficient was between RSD2 (computer) and RSD4 (mathematics activities with the computer) with $r = .70$ and $p < .0001$. The Pearson $r$ coefficient for the correlation between the TOLT and CTBS scores was statistically significant, $r = .298$, $p < .005$. This suggests a moderate relationship between students' TOLT and CTBS scores.

The Pearson $r$ coefficient for the intercorrelations between students' TOLT scores and their RSD concept scores ranged from zero ($r = .02$) to somewhat moderate ($r = .25$) and were positive in direction. Only two of the intercorrelations between TOLT and RSD concept scores were statistically significant. The correlation between TOLT and RSD5 (interaction with the mathematics teacher) scores and TOLT and RSD6 (interaction with the computer) scores were statistically significant with $p < .05$.

The Pearson $r$ coefficient for the intercorrelations between students' CTBS scores and their RSD concept scores ranged from zero ($r = .01$) to low ($r = .19$) in magnitude and were positive in direction. These intercorrelations were not statistically significant.

**Intercorrelations between CTBS and TOLT Scores and the Post-treatment RSD Scores**

These supplemental analyses were based on the Pearson
$r$ coefficient for the correlation between students' TOLT and CTBS scores and their post-treatment RSD scores. Table 16 presents the correlations between the CTBS and TOLT scores and the post-treatment RSD scores by treatment group (computer, textbook, and control).

The correlations between the different treatment groups TOLT scores ranged from zero ($r = .01$) to low ($r = .39$) in magnitude and were positive in direction. These correlations, with one exception, were not statistically significant. The only significant correlation was between the textbook group students' TOLT scores and their scores in the RSD concept "mathematics activities with the computer" ($r = .39, p < .0317$).

The correlations between the different treatment groups CTBS scores ranged from zero ($r = .01$) to low ($r = .34$) in magnitude and were positive in direction. These correlations were not statistically significant. The highest correlation was between the computer group students' TOLT scores and their scores in the RSD concept "interaction with the computer" ($r = .34, p > .0694$).

**RSD Subscales**

In order to reduce the number of concepts comprising the RSD scale used to measure students' affective level, two RSD subscales were considered using the total sums of
Table 16

**Pearson’s Intercorrelation Between CTBS and TOLT Scores and Post-treatment Scores**

<table>
<thead>
<tr>
<th>Instrument</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Computer Group (n = 89)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CTBS</td>
<td>.08</td>
<td>.21</td>
<td>.01</td>
<td>.23</td>
<td>.18</td>
<td>.34</td>
</tr>
<tr>
<td>TOLT</td>
<td>.01</td>
<td>.15</td>
<td>.23</td>
<td>.27</td>
<td>.16</td>
<td>.34</td>
</tr>
<tr>
<td><strong>Textbook Group (n = 31)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CTBS</td>
<td>.14</td>
<td>.16</td>
<td>.10</td>
<td>.24</td>
<td>.11</td>
<td>.21</td>
</tr>
<tr>
<td>TOLT</td>
<td>.06</td>
<td>.06</td>
<td>.13</td>
<td>.13</td>
<td>.39</td>
<td>.13</td>
</tr>
<tr>
<td><strong>Control Group (n = 30)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CTBS</td>
<td>.07</td>
<td>.26</td>
<td>.13</td>
<td>.03</td>
<td>.21</td>
<td>.29</td>
</tr>
<tr>
<td>TOLT</td>
<td>.03</td>
<td>.07</td>
<td>.07</td>
<td>.01</td>
<td>.14</td>
<td>.07</td>
</tr>
</tbody>
</table>

Note. RSD = Robustness Semantic Differential; RSD1 = mathematics; RSD2 = computer; RSD3 = mathematics activities; RSD4 = mathematics activities with the computer; RSD5 = interaction with the teacher; RSD = interaction with the computer; CTBS = Comprehensive Test of Basic Skills; TOLT = Test of Logical Thinking.
the three mathematics-related RSD concepts (mathematics teacher, mathematics activities, and interaction with the mathematics teacher) and three computer-related RSD concepts (computer, mathematics activities with the computer, and interaction with the computer). The RSD subscale-1 (RSDS-1) represents the sum of RSD1, RSD3, and RSD5 concept scores. The RSD subscale-2 (RSDS-2) represents the sum of RSD2, RSD4, and RSD6 concept scores. Table 17 presents the means and standard deviations of each RSD subscale by treatment group.

The structure of these RSD subscales is supported by the findings of a factor analysis of the pre-treatment RSD scores. Using students' total scores for each RSD concepts as the units of analysis, a factor analysis of the pre-treatment RSD data was employed. The factor analysis was an unconstrained varimax solution with orthogonal rotation and unity (1.0) as the estimate of communality. In this analysis, six factors were identified in the unrotated factor solution with two strong factors which together accounted for 71 percent of the total variance explained by the solution. These two components had eigenvalues 2.876070 and 1.377570. Two components were retained on the basis of the eigenvalues-greater-than-one rule, since the third eigenvalue was only 0.44814. In the rotated factor solution, these two factors had rotated eigenvalues
### Table 17

**Mean RSD Subscale Scores by Treatment Group**

<table>
<thead>
<tr>
<th>Treatment</th>
<th>RSD Subscale 1</th>
<th>RSD Subscale 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>86.17</td>
<td>127.39</td>
</tr>
<tr>
<td>SD</td>
<td>23.14</td>
<td>17.11</td>
</tr>
<tr>
<td>Z_Max</td>
<td>57.45</td>
<td>84.50</td>
</tr>
<tr>
<td>Textbook</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>83.45</td>
<td>126.35</td>
</tr>
<tr>
<td>SD</td>
<td>27.45</td>
<td>21.75</td>
</tr>
<tr>
<td>Z_Max</td>
<td>55.63</td>
<td>84.42</td>
</tr>
<tr>
<td>Control</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>88.63</td>
<td>123.63</td>
</tr>
<tr>
<td>SD</td>
<td>17.00</td>
<td>18.97</td>
</tr>
<tr>
<td>Z_Max</td>
<td>59.09</td>
<td>82.42</td>
</tr>
</tbody>
</table>

**Note.** RSD = Robustness Semantic Differential; RSD Subscale-1 = RSD1 + RSD3 + RSD5; RSD Subscale-2 = RSD2 + RSD4 + RSD6. Maximum Possible Score = 150. Z Max. = M / Maximum Possible Score.
2.214708 and 2.038941. The first component had large positive loadings for three RSD concepts (RSD2, RSD4 and RSD6). The second component had large loadings for the other three RSD concepts, with an especially high correlation for RSD3. The final communality estimates show that all the RSD concepts are well accounted for by the two components, with final communality estimates ranging from 0.752293 to 0.835862 and a final communality estimate total of 4.554149. Table 18 presents the rotated factor pattern for the RSD concept scores and final communality estimates.

Correlation Between CTBS and TOLT Scores and RSD Subscale Scores. These supplemental analyses were based on the Pearson r coefficients for the correlation between treatment groups CTBS and TOLT scores and their RSD subscale scores. Table 19 presents the correlation between CTBS and TOLT scores and RSD subscale scores by treatment group (computer, textbook, and control). The correlations ranged from very low \( r = .09 \) to moderate \( r = .34 \) in magnitude and were positive in direction. These correlations were not statistically significant. The highest correlation was between CTBS and RSDS-2 scores \( r = .34 \).

In general, the correlation between the textbook group students' CTBS and TOLT scores and their RSD
Table 18

**Rotated Factor Pattern Loading for RSD Concepts and Final Communality Estimate**

<table>
<thead>
<tr>
<th>Concept</th>
<th>Factor 1</th>
<th>Factor 2</th>
<th>FCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSD1</td>
<td>.12984</td>
<td>.82336</td>
<td>.698776</td>
</tr>
<tr>
<td>RSD2</td>
<td>.05420</td>
<td>.06413</td>
<td>.733765</td>
</tr>
<tr>
<td>RSD3</td>
<td>.08125</td>
<td>.88267</td>
<td>.785713</td>
</tr>
<tr>
<td>RSD4</td>
<td>.87074</td>
<td>.20514</td>
<td>.835862</td>
</tr>
<tr>
<td>RSD5</td>
<td>.23936</td>
<td>.70282</td>
<td>.751243</td>
</tr>
<tr>
<td>RSD6</td>
<td>.78136</td>
<td>.20436</td>
<td>.752290</td>
</tr>
</tbody>
</table>

**Note.** RSD = Robustness Semantic Differential; RSD1 = mathematics; RSD2 = computer; RSD3 = mathematics activities; RSD4 = mathematics activities with the computer; RSD5 = interaction with the teacher; RSD6 = interaction with the computer; FCE = Final Communality Estimate.
Table 19

Pearson $r$ Intercorrelation Between CTBS and TOLT RSD Subscale Scores

<table>
<thead>
<tr>
<th>Instrument</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer Group ($n = 28$)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CTBS</td>
<td>.09</td>
<td>.34</td>
</tr>
<tr>
<td>TOLT</td>
<td>.15</td>
<td>.09</td>
</tr>
<tr>
<td>Textbook Group ($n = 31$)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CTBS</td>
<td>.06</td>
<td>.18</td>
</tr>
<tr>
<td>TOLT</td>
<td>.04</td>
<td>.15</td>
</tr>
<tr>
<td>Control Group ($n = 30$)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CTBS</td>
<td>.15</td>
<td>.04</td>
</tr>
<tr>
<td>TOLT</td>
<td>.08</td>
<td>.39</td>
</tr>
<tr>
<td>Total Sample ($n = 89$)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CTBS</td>
<td>.06</td>
<td>.16</td>
</tr>
<tr>
<td>TOLT</td>
<td>.09</td>
<td>.19</td>
</tr>
</tbody>
</table>

Note. RSD = Robustness Semantic Differential; CTBS = Comprehensive Test of Basic Skills; TOLT = Test of Logical Thinking; RSD Subscale-1 = RSD1 + RSD3 + RSD5; RSD Subscale-2 = RSD2 + RSD4 + RSD6.
subscales scores was low. The highest correlation was between the CTBS and RSDS-2 scores with \( r = .18 \). The correlations between the control group students' CTBS and TOLT scores and their RSD subscale scores ranged from low \( (r = .04) \) to moderate \( (r = .39) \) in magnitude and were positive in direction. The highest correlation was between TOLT and RSDS-2 scores, with \( r = .39 \) and \( p < .0336 \).

Table 20 presents the Pearson \( r \) correlation coefficient between the posttest and retention test scores and RSD subscale scores by treatment group. In general, these correlations were moderate in magnitude and positive in direction for the different treatment groups. There were five statistically significant correlations. The highest correlation was between the textbook group students' retention test scores and their RSDS-1 scores \( (r = .46, p < .0085) \). The lowest correlation was between the control group students' posttest scores and their RSDS-1 scores.

**Multiple Regressions**

Multiple regression analyses were completed by regressing the dependent variables (students' posttest and retention test scores) on the set of independent variables (TDTL, CTBS, and RSD subscales) for each treatment group and for the total sample. These analyses were employed to
Table 20

**Pearson r Correlation Between Posttest and Retention Test Scores and RSD Subscale Scores**

<table>
<thead>
<tr>
<th>Instrument</th>
<th>Posttest</th>
<th>Retention Test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>UCVI</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Computer Group (n = 28)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RSDS-1</td>
<td>.33</td>
<td>.0072</td>
</tr>
<tr>
<td>RSDS-2</td>
<td>.38</td>
<td>.0456*</td>
</tr>
</tbody>
</table>

|                |          |                |
| **Textbook Group (n = 31)** |          |                |
| RSDS-1         | .31      | .0878          |
| RSDS-2         | .38      | .0324*         |

|                |          |                |
| **Control Group (n = 30)** |          |                |
| RSDS-1         | .26      | .1575          |
| RSDS-2         | .34      | .0585          |

|                |          |                |
| **Total Sample (n = 89)** |          |                |
| RSDS-1         | .27      | .0104          |
| RSDS-2         | .37      | .0003***       |

* * * p < .05. ** p < .01. *** p < .001.

**Note.** RSD = Robustness Semantic Differential; RSD Subscale-1 = RSD1 + RSD3 + RSD5; RSD Subscale-2 = RSD2 + RSD4 + RSD6; UCVI = Und. of the Concept of Variable Inst.
explore whether various combinations of independent variables would account for significant amounts of variance in the dependent variables. Of the variety of multiple regression procedures available, the Maximum R-Square procedure was used (SAS Institute, 1985). This procedure builds a series of regression models at each step in the analysis of various combinations of the independent variables. It tries to find the best one-variable model, the best two-variable model, and so forth.

**Posttest Scores as the Dependent Variable.** The first analysis was completed for the set of independent variables (TOLT, CTBS, RSDS-1, and RSDS-2) using computer group students' UCVI posttest scores as the dependent variable. Table 21 contains the results of the analysis. The first variable to enter the regression equation (highest single correlate with the dependent variable) was scores on the TOLT. This variable accounted for 36.5 percent of the total variation between computer group students' posttest scores. A best two-variable model was represented by the combination of scores on the TOLT with CTBS scores. This two-variable model accounted for 53.4 percent of the total variance for the total UCVI variation. The results in Table 21 also show the best three- and four-variable model in the analysis. The best four-variable model accounted for 61.2 percent of UCVI
Table 21

Summary of Maximum R-Square Multiple Regression of Computer Group Posttest Scores on All Independent Variables

<table>
<thead>
<tr>
<th>Step</th>
<th>Variables in Model</th>
<th>$R^2$</th>
<th>$\Delta R^2$</th>
<th>$F$</th>
<th>$p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TOLT</td>
<td>.365</td>
<td></td>
<td>14.95</td>
<td>.0007</td>
</tr>
<tr>
<td>2</td>
<td>TOLT, CTBS</td>
<td>.534</td>
<td>.169</td>
<td>14.36</td>
<td>.0001</td>
</tr>
<tr>
<td>3</td>
<td>TOLT, CTBS, RSDS-1</td>
<td>.584</td>
<td>.050</td>
<td>11.26</td>
<td>.0001</td>
</tr>
<tr>
<td>4</td>
<td>TOLT, CTBS, RSDS-1</td>
<td>.612</td>
<td>.028</td>
<td>9.07</td>
<td>.0002</td>
</tr>
</tbody>
</table>

Note. RSD = Robustness Semantic Differential; RSD Subscale-1 = RSD1 + RSD3 + RSD5; RSD Subscale-2 = RSD2 + RSD4 + RSD6; TOLT = Test of Logical Thinking; CTBS = Comprehensive Test of Basic Skills; $\Delta R^2$ = Percentage change in $R^2$ at each step of the analysis.
variation.

The results of this analysis show that TOLT, among the independent variable set, is the most important predictor variable explaining variation in the computer group students' UCVI scores. Thus, TOLT is the more potent predictor of computer group students' performance in the UCVI after the treatment than either the CTBS or RSD subscales.

The second regression analysis was completed for the set of independent variable using textbook group students' posttest scores as the dependent variable. Table 22 contains the results of the analysis. The first variable to enter the regression equation was RSDS-1. This variable accounted for 15.3 percent of the total variation between textbook group students' posttest scores. A best two-variable model was represented by the combination of RSDS-1 with TOLT. This two-variable model accounted for 25.2 percent of the posttest total variation. The best four-variable model accounted for 31.3 percent of the total variation.

The results of this analysis show that RSDS-1, among the independent variable set, is the most important variable explaining variation in the textbook group students' posttest scores. Thus, RSDS-1 is the more potent predictor of textbook group students' performance in the UCVI given as a posttest than the other independent
Table 22

Summary of Maximum R-Square Multiple Regression of Textbook Group Posttest Scores on All Independent Variables

<table>
<thead>
<tr>
<th>Step</th>
<th>Variables in Model</th>
<th>R²</th>
<th>ΔR²</th>
<th>F</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RSDS-1</td>
<td>.153</td>
<td></td>
<td>5.05</td>
<td>.0324</td>
</tr>
<tr>
<td>2</td>
<td>RSDS-1, TOLT</td>
<td>.252</td>
<td>.100</td>
<td>4.77</td>
<td>.0165</td>
</tr>
<tr>
<td>3</td>
<td>RSDS-1, TOLT, CTBS</td>
<td>.288</td>
<td>.030</td>
<td>3.58</td>
<td>.0268</td>
</tr>
<tr>
<td>4</td>
<td>RSDS-1, TOLT, CTBS</td>
<td>.313</td>
<td>.030</td>
<td>2.91</td>
<td>.0408</td>
</tr>
</tbody>
</table>

Note. RSD = Robustness Semantic Differential; RSD Subscale-1 = RSD1 + RSD3 + RSD5; RSD Subscale-2 = RSD2 + RSD4 + RSD6; TOLT = Test of Logical Thinking; CTBS = Comprehensive Test of Basic Skills; ΔR² = Percentage change in R² at each step in the analysis.
variables included in the study.

The third regression analysis was completed for the set of independent variable using control group students' posttest scores as the dependent variable. Table 23 contains the results of the analysis. The first variable to enter the regression equation was CTBS. This variable accounted for 49.7 percent of the total variation between control group students' posttest scores. A best two-variable model was represented by the combination of CTBS with RSDS-2. This two-variable model accounted for 60 percent of the posttest total variation. The best four-variable model accounted for 62.9 percent of the posttest total variation.

The results of this analysis show that CTBS, among the independent variable set, is the most important variable explaining variation in the control group students' posttest scores. Thus, CTBS is the more potent predictor of control group students' performance in the UCVI given as a posttest than the other independent variables included in the study.

The next regression analysis was completed for the set of independent variable using all students' (n = 89) posttest scores as the dependent variable. Table 24 contains the results of the analysis. The first variable to enter the regression equation was CTBS, which accounted for 24.7 percent of the total variation between students'
Table 23

**Summary of Maximum R-Square Multiple Regression of Control Group Posttest Scores on All Independent Variables**

<table>
<thead>
<tr>
<th>Step</th>
<th>Variables in Model</th>
<th>R²</th>
<th>ΔR²</th>
<th>F</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CTBS</td>
<td>.497</td>
<td></td>
<td>27.68</td>
<td>.0001</td>
</tr>
<tr>
<td>2</td>
<td>CTBS, RSDS-2</td>
<td>.600</td>
<td>.203</td>
<td>20.23</td>
<td>.0001</td>
</tr>
<tr>
<td>3</td>
<td>CTBS, RSDS-2, TOLT</td>
<td>.617</td>
<td>.017</td>
<td>13.04</td>
<td>.0001</td>
</tr>
<tr>
<td>4</td>
<td>CTBS, RSDS-2, TOLT</td>
<td>.629</td>
<td>.012</td>
<td>10.58</td>
<td>.0001</td>
</tr>
</tbody>
</table>

**Note.** RSD = Robustness Semantic Differential; RSD Subscale-1 = RSD1 + RSD3 + RSD5; RSD Subscale-2 = RSD2 + RSD4 + RSD6; TOLT = Test of Logical Thinking; CTBS = Comprehensive Test of Basic Skills; ΔR² = Percentage change in R² at each step in the analysis.
Table 24

**Summary of Maximum R-Square Multiple Regression of All Students' Posttest Scores on All Independent Variables**

<table>
<thead>
<tr>
<th>Step</th>
<th>Variables in Model</th>
<th>$R^2$</th>
<th>$\Delta R^2$</th>
<th>F</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CTBS</td>
<td>.247</td>
<td></td>
<td>28.59</td>
<td>.0001</td>
</tr>
<tr>
<td>2</td>
<td>CTBS, RSDS-2</td>
<td>.345</td>
<td>.098</td>
<td>22.63</td>
<td>.0001</td>
</tr>
<tr>
<td>3</td>
<td>CTBS, RSDS-2, TOLT</td>
<td>.389</td>
<td>.044</td>
<td>14.75</td>
<td>.0001</td>
</tr>
<tr>
<td>4</td>
<td>CTBS, RSDS-2, TOLT</td>
<td>RSDS-1</td>
<td>.412</td>
<td>.012</td>
<td>14.75</td>
</tr>
</tbody>
</table>

**Note.** RSD = Robustness Semantic Differential; RSD Subscale-1 = RSD1 + RSD3 + RSD5; RSD Subscale-2 = RSD2 + RSD4 + RSD6; TOLT = Test of Logical Thinking; CTBS = Comprehensive Test of Basic Skills; $\Delta R^2$ = Percentage change in $R^2$ at each step in the analysis.
posttest scores. A best two-variable model was represented by the combination of CTBS with RSDS-2. This two-variable model accounted for 34.5 percent of the posttest total variation. The best four-variable model accounted for 41.2 percent of the total posttest variation.

The results of this analysis show that CTBS, among the independent variable set, is the most important variable explaining variation in the students' posttest scores. Thus, CTBS is the more potent predictor of students' performance in the UCVI given as a posttest than the other independent variables included in the study. It is interesting to note that students' perception of the robustness of the computer-related environment was the second independent variable to enter the regression equation, even before TOLT which is considered an important predictor of students' learning.

**Retention Test Scores as the Dependent Variable.** A fifth exploratory regression analysis was completed for the independent variable set using computer group students' retention test scores as the dependent variable. The results of the fifth regression analysis are presented in Table 25. The first variable to enter the regression was the CTBS. This variable explains 37.7 percent of the total variance in the computer group.
Table 25

Summary of Maximum R-Square Multiple Regression of Computer Groups Retention Test Scores on All Independent Variables

<table>
<thead>
<tr>
<th>Step</th>
<th>Variables in Model</th>
<th>$R^2$</th>
<th>$\Delta R^2$</th>
<th>F</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CTBS</td>
<td>.377</td>
<td></td>
<td>15.77</td>
<td>.0005</td>
</tr>
<tr>
<td>2</td>
<td>CTBS, TOLT</td>
<td>.473</td>
<td>.096</td>
<td>14.78</td>
<td>.0003</td>
</tr>
<tr>
<td>3</td>
<td>CTBS, TOLT, RSDS-2</td>
<td>.548</td>
<td>.075</td>
<td>9.71</td>
<td>.0002</td>
</tr>
<tr>
<td>4</td>
<td>CTBS, TOLT, RSDS-2, RSDS-1</td>
<td>.574</td>
<td>.026</td>
<td>7.76</td>
<td>.0004</td>
</tr>
</tbody>
</table>

Note. RBD = Robustness Semantic Differential; RSD Subscale-1 = RSD1 + RSD3 + RSD5; RSD Subscale-2 = RSD2 + RSD4 + RSD6; TOLT = Test of Logical Thinking; CTBS = Comprehensive Test of Basic Skills; $\Delta R^2$ = Percentage change in $R^2$ at each step in the analysis.
retention test used as the dependent variable. The best two-variable regression model was CTBS and TOLT which accounted for 47.3 percent of the total variation. The best three-variable model accounted for 54.8 percent of the variation in the computer group retention test scores and consisted of, in order of their importance, the following variables: CTBS, TOLT, RSDS-2, and RSDS-1. These results suggest that CTBS is the most potent predictor of the computer group performance in the retention test than the other variables.

The following regression analysis was completed for the set of independent variables using textbook group students' retention test scores as the dependent variable. Table 26 contains the results of the analysis. The first variable to enter the regression equation was RSDS-1, which accounted for 21.5 percent of the total variation between textbook group students' retention test scores. A best two-variable model was represented by the combination of RSDS-1 with CTBS. This two-variable model accounted for 37.7 percent of the total variation. The best four-variable model accounted for 43.2 percent of the total variation.

The results of this analysis show that RSDS-1, among the independent variable set, is the most important variable explaining variation in the textbook group students' retention test scores. Thus, RSDS-1 is the most
Table 26

*Summary of Maximum R-Square Multiple Regression of Textbook Groups' Retention Test Scores on All Independent Variables*

<table>
<thead>
<tr>
<th>Step</th>
<th>Variables in Model</th>
<th>$R^2$</th>
<th>$\Delta R^2$</th>
<th>$F$</th>
<th>$p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RSDS-1</td>
<td>.215</td>
<td></td>
<td>7.97</td>
<td>.0085</td>
</tr>
<tr>
<td>2</td>
<td>RSDS-1, CTBS</td>
<td>.377</td>
<td>.162</td>
<td>8.48</td>
<td>.0013</td>
</tr>
<tr>
<td>3</td>
<td>RSDS-1, CTBS, TOLT</td>
<td>.427</td>
<td>.150</td>
<td>6.72</td>
<td>.0016</td>
</tr>
<tr>
<td>4</td>
<td>RSDS-1, CTBS, TOLT</td>
<td>.432</td>
<td>.005</td>
<td>4.95</td>
<td>.0042</td>
</tr>
</tbody>
</table>

Note. RSD = Robustness Semantic Differential; RSD Subscale-1 = RSD1 + RSD3 + RSD5; RSD Subscale-2 = RSD2 + RSD4 + RSD6; TOLT = Test of Logical Thinking; CTBS = Comprehensive Test of Basic Skills; $\Delta R^2$ = Percentage change in $R^2$ at each step in the analysis.
Table 27

Summary of Maximum R-Square Multiple Regression of Control Group Retention Test Scores on All Independent Variables

<table>
<thead>
<tr>
<th>Step</th>
<th>Variables in Model</th>
<th>R²</th>
<th>ΔR²</th>
<th>F</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CTBS</td>
<td>.463</td>
<td></td>
<td>24.35</td>
<td>.0001</td>
</tr>
<tr>
<td>2</td>
<td>CTBS, RSDS-2</td>
<td>.554</td>
<td>.111</td>
<td>17.17</td>
<td>.0001</td>
</tr>
<tr>
<td>3</td>
<td>CTBS, RSDS-2, TOLT</td>
<td>.596</td>
<td>.042</td>
<td>12.79</td>
<td>.0001</td>
</tr>
<tr>
<td>4</td>
<td>CTBS, RSDS-2, TOLT</td>
<td>.615</td>
<td>.029</td>
<td>9.95</td>
<td>.0001</td>
</tr>
</tbody>
</table>

Note. RSD = Robustness Semantic Differential; RSD Subscale-1 = RSD1 + RSD3 + RSD5; RSD Subscale-2 = RSD2 + RSD4 + RSD6; TOLT = Test of Logical Thinking; CTBS = Comprehensive Test of Basic Skills; ΔR² = Percentage change in R² at each step in the analysis.
potent predictor of textbook group performance in the UCVI given as a retention test than the other independent variables included in the study.

The seventh regression analysis was completed for the set of independent variables using control group students' retention test scores as dependent variable. Table 27 contains the results of the analysis. The first variable to enter the regression equation was CTBS. This variable accounted for 46.3 percent of the total variation between control group students' retention test scores. A best two-variable model was represented by the combination of CTBS with RSDS-2. This two-variable model accounted for 55.4 percent of the total variation. The best four-variable model accounted for 61.5 percent of the total variation.

The results of this analysis show that CTBS, among the independent variable set, is the most important variable explaining variation in the control group students' retention test scores. Thus, CTBS is the most potent predictor of control group students' performance in the UCVI given as a retention test than the other independent variables included in the study.

The final regression analysis was completed for the set of independent variables using all students' (n = 89) retention test scores as the dependent variable. Table 28 contains the results of the analysis. The first
Table 28

Summary of Maximum R-Square Multiple Regression of All Students' Retention Test Scores on All Independent Variables

<table>
<thead>
<tr>
<th>Step</th>
<th>Variables in Model</th>
<th>$R^2$</th>
<th>$R^2$</th>
<th>F</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CTBS</td>
<td>.246</td>
<td></td>
<td>28.11</td>
<td>.0001</td>
</tr>
<tr>
<td>2</td>
<td>CTBS, RSDS-2</td>
<td>.323</td>
<td>.077</td>
<td>20.61</td>
<td>.0001</td>
</tr>
<tr>
<td>3</td>
<td>CTBS, RSDS-2, TOLT</td>
<td>.351</td>
<td>.028</td>
<td>15.56</td>
<td>.0001</td>
</tr>
<tr>
<td>4</td>
<td>CTBS, RSDS-2, TOLT</td>
<td>.369</td>
<td>.018</td>
<td>16.20</td>
<td>.0001</td>
</tr>
</tbody>
</table>

Note. RSD = Robustness Semantic Differential; RSD Subscale-1 = RSD1 + RSD3 + RSD5; RSD Subscale-2 = RSD2 + RSD4 + RSD6; TOLT = Test of Logical Thinking; CTBS = Comprehensive Test of Basic Skills; $R^2$ = Percentage change in $R^2$ at each step in the analysis.
variable to enter the regression equation was CTBS. This variable accounted for 24.6 percent of the total variation between students' retention test scores. A best two-variable model was represented by the combination of CTBS with RSDS-2. This two-variable model accounted for 32.3 percent of the total variation. The best four-variable model accounted for 36.9 percent of the total variation. The results of this analysis show that CTBS, among the independent variable set, is the most important variable explaining variation in students' retention test scores. Thus, CTBS is the most potent predictor of students' performance in the UCVI given as a retention test than the other independent variables included in the study.

Reliability Analyses

Understanding of the Concept of Variable Instrument. In the present study, the Cronbach alpha reliability coefficient was computed for the control group students' \((n = 30)\) UCVI posttest scores. Control group students' scores in the posttest provided an unbiased estimate of UCVI reliability, the coefficient for all items being .87.

The UCVI included 32 items scored dichotomously. For the computer group, the UCVI posttest scores ranged from 4 to 26. For the textbook group, the scores ranged from 4
to 25. For the control group, the scores ranged from 2 to 18.

On the retention test, the UCVI scores ranged from 6 to 25 for the computer group, from 3 to 21 for the textbook group, and from 2 to 17 for the control group. In addition, the Pearson r coefficient for the correlation between the control group students’ posttest and retention test (given three weeks after treatment) scores was calculated as a measure of the instrument’s stability. The relationship between the posttest and retention test scores was strong ($r = .82$), statistically significant ($p < .0001$), and positive in direction. This suggests a high stability of the UCVI scores over time.

**Robustness Semantic Differential Concepts.** In the present study, Cronbach alpha coefficients were calculated for the pre-treatment RSD scores ($n = 89$). Reliability data for the six RSD concepts used in the study can be found in Table 29. In scoring RSD concept scores, odd numbered items were reversed scores (from 0 to 5). The reliabilities ranged from .83 to .92, with the lowest reliability being the RSD concept "mathematics teacher" (Alpha = .83) and the highest being "mathematics learning activities" (Alpha = .92).

In addition, the Pearson r coefficient for the correlations between pre-treatment RSD and post-treatment
Table 29

*Cronbach_Alpha_Reliability_Coefficients_for_the_RSD* Concepts

<table>
<thead>
<tr>
<th>RSD Concepts</th>
<th>Alpha Reliability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mathematics teacher</td>
<td>.83</td>
</tr>
<tr>
<td>Computer</td>
<td>.90</td>
</tr>
<tr>
<td>Mathematics learning activities</td>
<td>.92</td>
</tr>
<tr>
<td>Mathematics learning activities</td>
<td>.91</td>
</tr>
<tr>
<td>with the computer</td>
<td></td>
</tr>
<tr>
<td>Interaction with the mathematics</td>
<td>.88</td>
</tr>
<tr>
<td>teacher</td>
<td></td>
</tr>
<tr>
<td>Interaction with the computer</td>
<td>.91</td>
</tr>
</tbody>
</table>

*Note. RSD = Robustness Semantic Differential.*
RSD concept scores was calculated for the control group students. The control group RSD scores provided an unbiased estimate of scale stability. The correlation between students' scores in the pre- and post-treatment RSD1 was .78, pre- and post-treatment RSD2 was .82, pre- and post-treatment RSD3 was .78, pre- and post-treatment RSD4 was .88, pre- and post-treatment RSD5 was .82, and pre- and post-treatment RSD6 was .87. These results suggest the stability of the RSD concept scores over time.

**Suggested Model to Increase R-Squared**

Supplemental analyses were employed to improve the appropriateness of models using treatment as the main effect and students' posttest and retention test scores as dependent variables (see Appendix G). This model accounted for .093 of the total variation in posttest scores and .193 of the total variation in retention test scores. Several models were tried in an attempt to increase the R-Squared values. School and sex were used as blocking variables, and students' TOLT, CTBS and RSD scores were used as covariates. The interactions between these variables were not significant so they were pooled with the error term, and only the main effects were maintained.

The model that accounted for the highest total
variation of both students' posttest and retention test measures used treatment, school, sex, and CTBS as main effects. The R-Squared values for this model were .549 for the posttest and .623 for the retention test. The blocking variable of sex demonstrated greater statistical significance for the retention test measure than for the posttest measure. The results of these analyses can be found in Tables 30 and 31. Duncan's multiple-range test was employed to analyze the school and sex mean differences. There was a significant difference between LSU Lab School students' mean score (11.34) and Glen Oaks School students' mean score (5.78) on the posttest ($p < .01$) and LSU Lab School students' mean score (12.07) and Glen Oaks School students' mean score (6.32) on the retention test ($p < .01$). Also, there was a significant difference between male students' mean score (8.54) and female students' mean score (6.74) on the posttest ($p < .05$) and male students' mean score (9.38) and female students' mean score (7.13) on the retention test ($p < .05$).
Table 30

**Analysis of Variance for Suggested Model**

Dependent variable: Posttest Score for Understanding of the Concept of Variable

<table>
<thead>
<tr>
<th>Source of Variance</th>
<th>df</th>
<th>Sum of Squares</th>
<th>Mean Square</th>
<th>F</th>
<th>p</th>
<th>R²</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>5</td>
<td>949.93</td>
<td>189.99</td>
<td>16.86</td>
<td>.0001***</td>
<td>.504</td>
</tr>
<tr>
<td>Error</td>
<td>83</td>
<td>935.51</td>
<td>11.27</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Corrected Total</td>
<td>88</td>
<td>1885.44</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source of Variance</th>
<th>df</th>
<th>Sum of Squares</th>
<th>F</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Treatment</td>
<td>2</td>
<td>174.80</td>
<td>7.75</td>
<td>.0008***</td>
</tr>
<tr>
<td>School</td>
<td>1</td>
<td>654.31</td>
<td>58.05</td>
<td>.0001***</td>
</tr>
<tr>
<td>Sex</td>
<td>1</td>
<td>65.74</td>
<td>5.83</td>
<td>.0179*</td>
</tr>
<tr>
<td>CTBS</td>
<td>1</td>
<td>55.08</td>
<td>5.89</td>
<td>.0298*</td>
</tr>
</tbody>
</table>

* p < .05.  ** p < .01.  *** p < .001.
Table 31

Analysis of Variance for Suggested Model

Dependent variable: Retention Test Score for Understanding of the Concept of Variable

<table>
<thead>
<tr>
<th>Source of Variance</th>
<th>df</th>
<th>Sum of Squares</th>
<th>Mean Square</th>
<th>F</th>
<th>p</th>
<th>R²</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>5</td>
<td>1305.69</td>
<td>261.13</td>
<td>22.44</td>
<td>.0001***</td>
<td>.575</td>
</tr>
<tr>
<td>Error</td>
<td>83</td>
<td>966.07</td>
<td>11.63</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Corrected Total</td>
<td>88</td>
<td>2271.76</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source of Variance</th>
<th>df</th>
<th>Sum of Squares</th>
<th>F</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Treatment</td>
<td>2</td>
<td>438.17</td>
<td>18.82</td>
<td>.0001***</td>
</tr>
<tr>
<td>School</td>
<td>1</td>
<td>668.61</td>
<td>57.44</td>
<td>.0001***</td>
</tr>
<tr>
<td>Sex</td>
<td>1</td>
<td>126.92</td>
<td>10.90</td>
<td>.0014**</td>
</tr>
<tr>
<td>CTBS</td>
<td>1</td>
<td>71.98</td>
<td>6.18</td>
<td>.0149*</td>
</tr>
</tbody>
</table>

* p < .05. ** p < .01. *** p < .001.
CHAPTER 5
Summary, Discussion, and Recommendations

The main objective of this study was to investigate whether there was a significant difference in understanding the concept of variable and in attitudes among sixth grade students who used a computer programming approach, students who used a textbook-based approach, and students who received no instruction. In addition, the strength of the relationship between students' posttest and retention test scores and their pre-treatment scores was analyzed. A discussion of the results presented in the preceding chapter is provided in the following sections.

Effects of Treatment on Students' Understanding of the Concept of Variable at the End of the Experiment

The first hypothesis of the study stated that students receiving computer programming instruction would demonstrate greater understanding of the concept variable on the posttest than students receiving textbook-based instruction. The data did not support the predictive hypothesis. There was no significant difference between posttest scores of students receiving programming instruction and students receiving textbook-based instruction on the concept, although the computer
programming group mean was greater than the textbook group. This finding indicates a comparable treatment effect on students' understanding of variable. Students receiving programming instruction might have the advantage of acquiring two skills, computer programming and mathematics, a point consistent with Jordan's research (1985).

The second hypothesis of the study stated that students receiving computer programming instruction would demonstrate greater understanding of the concept of variable on the posttest than students receiving no instruction on the concept of variable. There was a significant difference in understanding the concept at the end of the treatment in favor of the computer programming group. The students who received programming instruction had a higher mean score than students who did not receive instruction on the concept. The investigator considered the difference of moderate educational significance. The computer programming instruction provided a model which was more effective than no instruction at all. Because of this lack of instruction, students in the control group were unfamiliar with the concept of variable and did not have the opportunity to learn the concept.

The third hypothesis in this study stated that students receiving textbook-based instruction would demonstrate greater understanding of the concept of
variable on the posttest than students receiving no instruction on the concept of variable. There was no significant difference between students receiving a textbook-based approach and students receiving no instruction on the concept. The data did not support this hypothesis. Although the difference was not a significant, the students receiving textbook-based instruction on variable produced a higher mean score on the posttest than students receiving no instruction.

It is possible that the textbook-based approach did not provide an effective model of the concept, and hence did not meet the educational needs of the sixth grade students. Sixth grade students need concrete props and applications that facilitate their learning. Students receiving textbook-based instruction scored no higher on the concept of variable test than students in the control group. Improvements need to be made in order to make this approach more effective. One alternative could be to combine or incorporate other concrete models of the concept to the textbook-based approach presented in this study. The textbook-based approach provided practice and repetition of the mathematical ideas such as expression and equation, but apparently this technique was not enough to make a difference.
Effects of Treatment on Students' Understanding of the Concept of Variable Three Weeks after Treatment

The fourth hypothesis stated that students receiving programming instruction would demonstrate greater understanding of the concept of variable on the retention test (three weeks after treatment) than students receiving textbook-based instruction. There was a significant difference in understanding the concept in favor of the students receiving computer programming instruction. The students who received computer programming instruction had a higher mean posttest score than students receiving textbook-based instruction.

Computer programming group students appear to have retained their understanding of variable better than the textbook group students. A possible explanation relates to the strength of the effect computer programming instruction has on students' understanding of the concept. The computer programming approach provided an effective model for teaching variable and one which seems more appropriate than the textbook-based approach. In Bloom's (1976) terms, the computer programming approach provided for improving the quality of instruction. Some sixth grade students are capable of formal reasoning but they can still profit from concrete activities, such as computer programming activities based on their conceptual...
orientation (Inhelder & Piaget, 1958; Piaget, 1976). LOGO graphics provide a way to visualize how changing the values of variable affects the design on the computer monitor. The idea that computer programming enhances students' understanding of mathematics is supported by other studies in this area (Washburn, 1970; Feurzeigh et al.; Oprea, 1985).

The fifth hypothesis stated that students receiving programming instruction would demonstrate greater understanding of the concept of variable on the retention test than students receiving no instruction on the concept. There was a significant difference in understanding of variable in favor of the students receiving computer programming instruction. The students who received computer programming instruction had a higher mean retention test score than students who did not receive instruction.

A possible explanation for this finding is that the programming approach allowed and encouraged the students to learn the concept, but students in the control group did not have this opportunity and did not understand the as well as the computer group.

It its interesting to note that retention test means for the computer and control group were higher than their posttest score means. This suggests an interaction between testing effects and learning. Possibly, however,
learning experiences between each administration of the posttest and retention test might have affected the computer and control group students' learning in some ways.

The sixth hypothesis stated that students receiving textbook-based instruction would demonstrate greater understanding of the concept of variable on the retention test (three weeks after treatment) than students receiving no instruction on the concept of variable (control group students). The data did not support this hypothesis. There were no significant differences between the students receiving the textbook-based instruction and students receiving no instruction three weeks after treatment.

One reason for this finding may be that the textbook-based approach did not provide for sixth grade students' needs, such as mental models to deal with abstract ideas in order to understand the concept. LOGO computer programming provides a concrete model for variable which is not provided in the textbook-based approach. This model includes concrete props and visual models that might help the students' learning of the concept. This is more relevant because several studies have found that students have various misconceptions of the concept (Wagner, 1981; Clement, 1982).

The LOGO computer programming activities used in this study emphasized the application of variable as it is used.
in mathematics. The concept was used as a symbol that may be replaced by any member of a given set of numbers. In this setting each variable corresponds to a distinct location in the memory of the computer. This provides concrete props and applications of the concept that might have enhanced students' learning and retention of learning. When students write, run, and edit different LOGO programs involving variables, they are involved in concrete and motivational applications of the concept. Analyzing and observing LOGO graphic programs might have provided an effective and easy to remember (long-term memory) visual model which is appropriate for their cognitive development.

In addition, the programming instruction facilitated more active involvement on the part of students. This active involvement might have also provided for students' gains in understanding. This supports Bloom's (1976) claim that time on task is an important factor in the teaching-learning process.

Effects of Treatment on Students' Attitudes Toward Mathematics at the End of the Treatment

The seventh hypothesis of the study stated that students receiving programming instruction would have more positive attitudes toward mathematics at the end of the
treatment. The results did not support this hypothesis. There was no significant difference between treatment groups' attitudes toward mathematics as measured by the RSD administered at the end of the treatment. The RSD included six target objects, and no significant differences were found in terms of students' attitudes toward any of these target objects. Students' attitudes toward the different target objects were consistent from one treatment group to the other.

Students' attitudes toward the mathematics teacher, mathematics activities, and interaction with the teacher were found to be low. In general, students perceived the mathematics environment as nondynamic, unimportant, dull, and boring at the end of the treatment. This attitude was consistent with the one students had at the start of the study. Possible explanations for this finding are a lack of motivation and challenge in the mathematics classes and the students' view of the teacher as rigid and sometimes negative (Licata and Wildes, 1980). These suggest a need for increasing the motivational aspect of the learning environment.

Students' attitudes toward the computer, mathematics activities with the computer, and interaction with the computer were found to be positive and moderately high. They generally perceived these target objects as interesting, challenging, active, powerful, thrilling,
important, fresh, meaningful, and action-packed. There exist several possible reasons: the computer provided spontaneous student involvement (on many occasions students would continue working with the computer in the absence of the teacher); the computer seems to be a moving, dynamic focal point for the class; the computer class atmosphere appears to be informal with the students holding a degree of autonomy over the activities; and students see the mathematics activities with the computer as "fun" and look forward to working with it (Licata and Wildes, 1980). Lepper (1985) noted that interest in computers is generally high which make them very motivating for the students. The incorporation of computers into mathematics activities also seems to make mathematics very motivating and interesting to the students. According to Bloom (1976), students' affective entry is an important factor in terms of learning outcomes. Further research analyzing specific aspects of the use of computers to enhance the mathematics environment and students' attitudes toward mathematics is necessary.
Relationship Between Students' Pre-treatment Scores and Their Posttest and Retention Test Scores

The strength of the relationship between students' understanding of the concept of variable posttest and retention test scores and their pre-treatment scores in mathematics achievement, logical thinking, and affective level was analyzed by treatment group. There was no significant relationship between the computer programming students' posttest and retention test scores and their scores on each of the RSD subscales. The same was true for the textbook and control group students. Students' attitudes appeared to be unrelated to computer, textbook, and control group students' performance on either the posttest or retention test. One reason may be that the students had a consistent attitude toward the different target objects included in the RSD scale. Students' attitudes toward the computer, the mathematics activities with the computer, and the interaction with the computer were consistently high. Similarly, students' attitudes toward the mathematics teacher, the mathematics activities, and the interaction with the mathematics teacher were consistently low. It is possible that the unit of analysis might have affected the results of the study. Morris (1986) found a strong relationship between RSD concepts (as perceived by teachers) and students'
achievement (as measured by the Comprehensive Assessment Program).

There was a significant positive relationship between the computer programming group students' posttest scores and their scores in the achievement test. This was also true for the relationship between textbook group and control group students' posttest scores and their achievement test scores. A similar relationship was found between each treatment group (computer, textbook, and control) retention test scores and their scores in the achievement test.

These findings suggest that students' previous mathematics achievement may be involved in students' performance in the different treatment activities and understanding of the concept at the end or three weeks after treatment. Students' entry mathematics achievement is a factor that needs closer attention and analysis in future research. It appears to be a significant predictor of sixth grade, novice programmers' success in learning of variable through programming activities and transferring programming concepts and skills to mathematics activities. The relationship seems apparent: the higher students' entry mathematics achievement, the greater the possibility of learning mathematics concepts and transferring programming skills to nonprogramming activities. This finding is in accordance with Papert, et al. (1979) who
considered students' entry achievement as the main factor affecting students' performance in different assessment tasks (for example, proportional thinking tasks, and conservation of number tasks).

Computer and textbook group students' posttest scores were also found to have a significant positive relationship with their logical thinking scores. The same positive relationship was found between computer and textbook group students' retention test scores and their scores in the logical thinking test. The relationship between control group students' posttest scores and their scores in the logical thinking test was not significant, but there was a significant positive relationship between control group students' retention test scores and their scores in the logical thinking test.

As with students' mathematics achievement scores, students' logical thinking scores may be related to students' understanding of the concept of variable after receiving computer programming or textbook-based instruction. Students' logical thinking seems to be an important factor in predicting students' success in transferring programming concepts and skills to nonprogramming activities. Further research is needed in this area to analyze the extent of this relationship.
Recommendations for Further Research

A number of recommendations for further research can be generated from the results of this study. Recommendations include changes in the design of the study, modifications of the instrument used to measure students' understanding of the concept of variable, and extension of the present research.

Some of the findings of this research were exploratory, and as in any research, there is a need for replication. The number of training sessions received by the students' could be lengthened to determine whether additional practice would result in greater gains in students' understanding of the concept of variable, and thus extend the findings of this research study. The number of students included in the sample could also be increased. More students would provide a more representative sample of the population and more conclusive results as to whether students' characteristics (cognitive and affective entries) are important factors in students' gains of understanding of variable at the of treatment.

The UCVI proved to be a reliable and valid measure of students' understanding of the concept. However, enhancements of and modifications to the instrument could be developed to support further research questions. Additional testing of the instrument is recommended. Some
items could be refined or eliminated from the test. Also, the addition of items to gather more specific data regarding students' understanding of the concept could provide a better indication of the specific treatment effects. The instrument does not reveal the thinking process used by the students in solving the problems and applying and transferring their understanding of the concept of variable. An analysis of individual students' protocols in solving UCVI problems or other similar questions would be a valuable complement to these findings.

The UCVI scores were found to be low for many students. Although the instrument was found to be reliable, it was not sensitive enough to acknowledge gains in learning. The items might have been too difficult or have required a higher developmental level than the students possessed. In other words, the students needed to think more abstractly than what they were capable of doing. The test would be more effective with students of a higher logical thinking level. The TOLT scores showed that most students were not able to reason on an abstract level. Using a more sensitive and developmentally appropriate measure might have reflected higher differences between treatment groups' learning. The UCVI low scores also suggest that the concept might indeed be too abstract and difficult to learn for sixth graders.
Normative data is needed in order to analyze differences among students' understanding of the concept of variable at different developmental and grade levels. The replication of the study using students with higher developmental and grade (seventh or eighth) levels is recommended.

The TOLT was an effective measure of students' logical thinking level. The low correlation between TOLT and CTBS scores suggests that the TOLT measures of logical thinking are independent of school matter (mathematics achievement in this case). TOLT seems to indeed measure logical thinking as suggested by Tobin and Capie (1980).

The computer learning environment is more robust than the textbook-based learning environment. In the multiple regression analysis, students' robustness scores for the computer-related concepts enter as the second predictor of learning. This suggests that environmental robustness is an important construct predicting learning outcomes. Given the importance of environmental robustness and the positive effects of computers on environmental robustness, it is advisable to use computers in the mathematics teaching-learning process. More research analyzing the specific factors that makes the computer environment more robust and how the teachers can use this factor to increase the mathematics environment robustness is necessary.
Since students' cognitive (logical thinking and mathematics achievement) and affective entry were related to students' gains in understanding, more research in controlling these variables could provide better indications of specific differences among students, their understanding of the concept, and the effects of computer programming on students' understanding of the concept. Other researchers might want to investigate the relationship between students' understanding of the concept of variable after treatment and other learner characteristics, such as reading level and cognitive style.

Results from Milner (1973) with fifth grade students, Oprea (1985) with sixth grade students, and Clement (1982) with college students suggest a relationship between programming ability and understanding of the concept of variable. The results of the present study appear to support these findings. Similar findings have been found with other specific mathematics skills and concepts such as equation, iteration, and function (Dwyer, 1975; Feurzeig et al., 1969; Krull, 1980; Lawler, 1980; Oprea, 1985; Papert, et al., 1979; Prichard, 1985; Soloway, Lochhead and Clement, 1982; Washburn, 1970). The LOGO computer approach used in the present study appears to be more effective than the textbook-based approach to introduce the concept of variable to sixth
grade students. The effectiveness of the LOGO computer programming approach appears to be stronger for helping students' long-term retention of the concept. The computer programming activities appear to have a delayed effect on certain aspects of achievement as was the case in Clements' research (1980). Further research on retention and delayed effects of computer programming is necessary in order to support these findings. Most studies in this area have not accounted for delayed or retention effects of computer programming activities. According to Byers and Earlwanger (1985), there seems to be a gap between the theory and practice of mathematics education and the developments in memory research. Investigators need to pay closer attention to the effectiveness of different models such as computer programming to enhance mathematics learning.

The present study's findings appear to support a recommendation by Corbitt (1985) who advocates an early introduction of algebraic concepts in elementary school through programming activities. Usually, the concept of variable is not formally introduced until eighth grade in pre-algebra or algebra classes, but the early use of programming activities, such as the ones presented in this study, might help students overcome and avoid later misconceptions of the concept of variable. Evidence of students' misconceptions of the concept of variable have
been reported by several investigators (Carpenter et al.,
1981; Clement, Lockhead and Monk, 1981; Rosnick, 1981;

The analysis of one specific programming skill, in
this case understanding of the concept of variable, was an
effective approach to investigate transfer of programming
skills to nonprogramming activities. It appears that
novice programmers (with limited programming training)
might be able to transfer their understanding of the
concept of variable in a computer programming setting to a
nonprogramming mathematics setting. This finding is
related to findings in the area of novice programmers’
transfer of specific computer programming skills to
nonprogramming activities (Billings, 1983; Hungate, 1982;

Based upon the results of this study, it appears to
be useful to teach sixth grade students the concept of
variable through computer programming activities.
Although no significant difference was found between the
textbook-based and the computer programming approach in
teaching the concept of variable, there was a significant
difference for retention test scores. The computer
activities presented a concrete approach that appeared to
provide for sixth grade students’ learning and long-term
retention of the concept. These findings provide a basis
for an early introduction of the concept of variable in
elementary school through computer programming activities. Furthermore, the activities used in this study appear to provide a medium by which this can be accomplished. The effectiveness of this approach seems to be related to students' cognitive and affective entries. Further research is necessary for a better understanding of the roles that students' mathematics achievement and logical thinking play in the instructional process.
REFERENCES


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


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


APPENDICES
APPENDIX A

Pre-treatment Instructional Lessons

Lesson 1 - Introducing the Turtle

Lesson Focus

To introduce the turtle and the turtle graphics world, introduce LOGO primitive commands, use shortcuts to write these primitive commands, and make the turtle draw designs.

Materials

LOGO on the classroom computers, copies of MAZE 1, MAZE 2, Activities 1, 2 and 3, pencils, protractor transparency, and chalkboard.

Group Size

Whole-class, and individual (or partners).

Time for Activities

Introduction 10 minutes (MAZE 1 and 2), each student at the computer for Activity-1 10 minutes, for Activity-2 15 minutes, and for Activity-3 15.

Teaching Notes

1. Load the LOGO language before students start working at the computers.

2. Tell the students that they are going to learn how to draw designs on the computer screen using a computer language called LOGO. Ask students to stand up and follow some commands you are going to give. This activity is to help students understand the four basic LOGO commands: FORWARD, BACK, RIGHT, and LEFT. Give the following commands to the students: FD 3 RT 90 BK 3 LT 90 RT 90 LT 180. Explain that the turtle turns around using degrees. Distribute copies of MAZE 1 to the students. Allow the students to work independently. Do the same for MAZE 2.

3. Distribute copies of Activity 1 and the protractor transparency to the students. Notice that this activity uses a guided discovery approach. Allow them to work independently at the computers for 15 minutes.
Make sure students write the turtle's messages. In exercise 5, they need to sketch what several commands do. Remind them to press the RETURN key each time they finish typing a line, and observe if the turtle moves or not after each command. Explain how to use the protractors to measure angles.

4. As they work, answer any question they might have. It is important they understand what each LOGO command is, and what it does. Make sure they understand that each command starts where the last one ended, and that forward can be in different directions depending on direction the turtle is facing when the command FORWARD is given. Sometimes it is difficult to determine in which direction the turtle is heading. Students should notice that the point is all white, and the base has a black area. The turtle will turn in the direction given by the commands RIGHT (clockwise) or LEFT (counterclockwise).

5. Help students understand that the computers will not check for mistakes. Students must work carefully and check themselves. Each time they enter a command in the computer it must be in the exact form specified, including correct spelling, spacing, and the use of special keys when necessary. Help students correct errors.

6. Distribute copies of Activities 2 and 3 to the students. Allow them work independently (or by partners) at the computers. These activities provide students with the opportunity to explore different LOGO primitive commands, including: DRAW, FORWARD, BACK, RIGHT, LEFT, PENUP, PENDOWN, SHOWTURTLE, HIDETURTLE and HOME.

7. Help students explore the primitive commands presented in these activities. HOME is an extremely useful command, especially when the turtle is out of range. It is the name of the original position of the turtle, in the center of the screen facing up. At any time, if they want the turtle to return to its home position, they just need to enter the word HOME. Everything on the screen will remain as is except for the turtle's location, plus the turtle leaves a trail from where it was to its home.

8. Help students observe that the primitive command PENUP or PU lifts the pen up so no drawing occurs, and that after they have located the turtle in the spot where they intend to resume drawing, they need to enter PENDOWN or PD. Only by typing PENDOWN or PD and pressing RETURN will the turtle begin drawing again.
Which of the following sequences tell how to get from B to E?

1. FD 7 RT 90 FD 5 LT 90 FD 16 RT 90 FD 15 RT 90 RT 90 FD 4 LT 90 FD 5 LT 90 FD 6

2. FD 5 RT 90 FD 5 LT 90 FD 16 LT 90 FD 15 RT 90 FD B RT 90 FD 4 LT 90 FD 5 LT 90 FD 6

3. FD 5 RT 90 FD 5 LT 90 FD 16 RT 90 FD 15 RT 90 FD B RT 90 FD 4 LT 90 FD 5 LT 90 FD 6
Write the commands needed to move the turtle from B to E.
Name ___________________________ Activity 1

Talking to the Turtle

1. You are going to learn how to draw designs on the computer screen using
turtle graphics. Type DRAW, then press RETURN and you will see the turtle.

2. Sketch what you see on the screen. This is a picture of the turtle and the turtle's
world. The turtle is at its HOME position. Note that it is facing or heading straight
up, in the center of the screen.

3. Type each of the following commands and press RETURN. Record the message
you see when you give each of the following commands to the turtle.

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>TURTLE'S MESSAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. MOVE</td>
<td>_______________________________</td>
</tr>
<tr>
<td>b. FORWARD</td>
<td>_______________________________</td>
</tr>
<tr>
<td>c. BACKWARD</td>
<td>_______________________________</td>
</tr>
<tr>
<td>d. BACK</td>
<td>_______________________________</td>
</tr>
<tr>
<td>e. TURN</td>
<td>_______________________________</td>
</tr>
<tr>
<td>f. RIGHT</td>
<td>_______________________________</td>
</tr>
<tr>
<td>g. LEFT</td>
<td>_______________________________</td>
</tr>
</tbody>
</table>

4. The turtle has a dictionary of words or commands called "primitives." The words MOVE, BACKWARD, and TURN are not in the turtle's dictionary. What message does the turtle give you when you give it a command that is not in its dictionary?

The commands FORWARD, BACK, RIGHT, and LEFT are in the turtle's dictionary, but these commands need more information. What message does the turtle give you when you give a command that needs more information?

5. Try these commands and sketch what you see the turtle draw after you
give all the commands. Observe how the turtle moves.

FORWARD 30 Sketch:
RIGHT 90
BACK 50
LEFT 120

Each tiny movement the turtle makes forward or back is called a turtle step. For example, the command FORWARD 10 indicates 10 turtle steps.

Each movement the turtle makes when it turns around to the right or left is called a degree. For example, the command RIGHT 90 indicates that the turtle turns around 90 degrees. See the protractor in next page.
Activity I continued ...

The chart below summarizes the keys and functions you will need when you are typing the different commands. The computer will not check for mistakes. You must work carefully and check and correct yourself. Each time you enter a command in the computer it must be in the exact form specified, including correct spelling, spacing and the use of special keys when necessary.

The notation SHIFT-N means to depress the SHIFT key and press the N key at the same time. The notation CONTROL-F is similar, but the CONTROL key is to be depressed while the F key is pressed.

<table>
<thead>
<tr>
<th>KEY(S) TO PRESS</th>
<th>TO DO</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPACE BAR</td>
<td>spacing</td>
</tr>
<tr>
<td>RETURN</td>
<td>return control to computer</td>
</tr>
<tr>
<td>ESC</td>
<td>going backward with eraser</td>
</tr>
<tr>
<td>&lt;----</td>
<td>going backward without eraser</td>
</tr>
<tr>
<td>-----)</td>
<td>going forward without eraser</td>
</tr>
<tr>
<td>CONTROL-F</td>
<td>fullscreen-uses whole screen to show graphics world</td>
</tr>
<tr>
<td>CONTROL-S</td>
<td>splitscreen-shows most of graphics world and 4 text lines</td>
</tr>
<tr>
<td>CONTROL-T</td>
<td>textscreen-use whole screen to show text lines</td>
</tr>
</tbody>
</table>
**Turtle Drawing**

The first set of primitive commands to explore contains:

<table>
<thead>
<tr>
<th>DRAW</th>
<th>FORWARD</th>
<th>BACK</th>
<th>RIGHT</th>
<th>LEFT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Help the turtle draw by giving the commands below. Remember to press RETURN after you type each line. In the space next to each list of commands, sketch what the turtle draws after you give all the commands to the turtle.

<table>
<thead>
<tr>
<th>1. DRAW Sketch:</th>
<th>2. DRAW Sketch:</th>
</tr>
</thead>
<tbody>
<tr>
<td>RIGHT 30</td>
<td>FORWARD 100</td>
</tr>
<tr>
<td>FORWARD 20</td>
<td>RIGHT 90</td>
</tr>
<tr>
<td>LEFT 60</td>
<td>FORWARD 100</td>
</tr>
<tr>
<td>BACK 20</td>
<td>RIGHT 90</td>
</tr>
<tr>
<td>RIGHT 60</td>
<td>FORWARD 100</td>
</tr>
<tr>
<td>FORWARD 20</td>
<td>RIGHT 90</td>
</tr>
<tr>
<td>LEFT 60</td>
<td>FORWARD 100</td>
</tr>
<tr>
<td>BACK 20</td>
<td>RIGHT 90</td>
</tr>
<tr>
<td>LEFT 60</td>
<td>FORWARD 100</td>
</tr>
<tr>
<td>FORWARD 40</td>
<td>RIGHT 90</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3. DRAW Sketch:</th>
<th>4. DRAW Sketch:</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORWARD 40</td>
<td>BACK 10</td>
</tr>
<tr>
<td>LEFT 120</td>
<td>RIGHT 90</td>
</tr>
<tr>
<td>FORWARD 40</td>
<td>FORWARD 50</td>
</tr>
<tr>
<td>LEFT 120</td>
<td>LEFT 90</td>
</tr>
<tr>
<td>FORWARD 40</td>
<td>FORWARD 10</td>
</tr>
<tr>
<td>LEFT 120</td>
<td>LEFT 90</td>
</tr>
<tr>
<td></td>
<td>FORWARD 50</td>
</tr>
<tr>
<td></td>
<td>RIGHT 90</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5. DRAW Sketch:</th>
<th>6. DRAW Sketch:</th>
</tr>
</thead>
<tbody>
<tr>
<td>LT 45</td>
<td>FD 30</td>
</tr>
<tr>
<td>FD 30</td>
<td>RT 135</td>
</tr>
<tr>
<td>BK 30</td>
<td>FD 40</td>
</tr>
<tr>
<td>RT 90</td>
<td>LT 135</td>
</tr>
<tr>
<td>FD 30</td>
<td>FD 30</td>
</tr>
<tr>
<td>BK 30</td>
<td></td>
</tr>
<tr>
<td>LT 45</td>
<td></td>
</tr>
<tr>
<td>BK 40</td>
<td></td>
</tr>
</tbody>
</table>
Name

Shortcuts

You can use shortcuts for these commands:

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>SHORT FORM</th>
<th>COMMAND</th>
<th>SHORT FORM</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORWARD</td>
<td>FD</td>
<td>PENDOWN</td>
<td>PD</td>
</tr>
<tr>
<td>BACK</td>
<td>BK</td>
<td>PENUP</td>
<td>PU</td>
</tr>
<tr>
<td>RIGTH</td>
<td>RT</td>
<td>SHOWTURTLE</td>
<td>ST</td>
</tr>
<tr>
<td>LEFT</td>
<td>LF</td>
<td>HIDETURTLE</td>
<td>HT</td>
</tr>
</tbody>
</table>

Help the turtle draw by giving the commands below. In the space next to each list of commands, sketch what the turtle draws after to give the list of commands.

Observe that the command PU lifts the turtle's pen up so no drawing occurs, and after you have located the turtle in the spot where you want to start drawing, you need to type PD and press RETURN. Only by typing PD and pressing RETURN will the turtle begin drawing again.

1. DRAW Sketch:
   - RT 90
   - PENUP
   - FD 20
   - PENDOWN
   - FD 20
   - PENUP
   - LT 90
   - FD 10
   - PENDOWN
   - FD 10
   - HOME

2. DRAW Sketch:
   - RT 90
   - PENUP
   - FD 20
   - PENDOWN
   - FD 20
   - PENUP
   - LT 90
   - FD 10
   - PENDOWN
   - FD 10
   - HOME

3. DRAW Sketch:
   - HIDETURTLE
   - BK 20
   - FD 20
   - LT 90
   - FD 20
   - BK 90
   - FD 20
   - LT 90
   - FD 20
   - PU
   - SHOWTURTLE
   - RT 45
   - FD 45
   - FD 10
   - PD
   - FD 10
   - PU
   - BK 30
   - PD
   - BK 10
   - HIDETURTLE

4. DRAW Sketch:
   - HT
   - PU
   - RT 90
   - PD
   - ST
   - PD
   - BK 10
   - HOME
   - ST
   - PU
   - BK 30
   - PD
   - BK 10
   - HIDETURTLE

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Lesson 2 - More Turtle World

Lesson Focus

To make the turtle draw designs, record the commands needed to draw a design, and sketch the designs the turtle draws.

Materials

LOGO on the computers, Activities 4 and 5, pencils, and chalkboard.

Group Size

Whole-class, and individuals (or partners).

Time for Activities

Whole class discussion 5 minutes, each student on the computers for Activity-4 20 minutes, and for Activity-5 25 minutes.

Teaching Notes

1. Load the LOGO language before students start working on the computers.

2. Review with the students the fact that they have been giving the turtle commands that make it draw on the screen. In this lesson they are going to have the opportunity to draw more designs using the turtle graphics and the commands they have learned so far.

3. Distribute copies of Activity 4 to the students. Allow them to work independently.

4. In exercise 1, they should draw (roughly) the resulting stair, illustrated below:

```
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>--</td>
</tr>
</tbody>
</table>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
5. Exercise 2 is a series of commands that will make a familiar object. Students should predict what each instruction will do before they enter the command. This is helpful in anticipating movements when they begin to design their own projects. The graphic they entered is number one (see the figure above). They might notice some distortion on their screen causing the corners of the lines to appear as if they were going beyond the bounds of the object. This occurs in some monitors and not on others. For now, tell the students not to worry about it as it does not indicate that their program is inaccurate.

```
CS
BK 40
FD 100
LT 90
FD 20
RT 135
FD 75
RT 135
FD 154
LT 90
FD 15
RT 90
FD 10
RT 90
FD 63
RT 90
FD 10
RT 90
FD 16
```

6. For exercises 3, 4, 5 and 6 of activity 4, use attribute blocks to show the students how the turtle move and turn. Allow them to use the attribute blocks as concrete representations.

7. Distribute copies of Activity 5 to the students. Allow them to work independently (or by partners) at the computers.
Turtle Designs

Type DRAW and press RETURN to find the turtle. So far you have been giving the turtle commands that make it draw on the screen. In this activity you are going to have the opportunity to draw more designs.

Use the commands below to do the following exercises.

| FORWARD (FD) | PENDOWN (PD) |
| BACK (BK) | PENUP (PU) |
| RIGHT (RT) | DRAW |
| LEFT (LT) | HOME |

Read the commands and sketch what you think the list of commands will tell the turtle to draw. Then check your prediction by giving the list of commands to the turtle. If your prediction was inaccurate, correct your sketch.

1. **DRAW**
   Sketch:
   - `FD 25`
   - `RT 90`
   - `FD 25`
   - `LT 90`
   - `FD 25`
   - `RT 90`
   - `FD 25`
   - `LT 90`
   - `FD 25`
   - `RT 90`
   - `FD 25`
   - `LT 90`

2. **DRAW**
   Sketch:
   - `BK 40`
   - `FD 100`
   - `LT 90`
   - `FD 20`
   - `RT 135`
   - `FD 75`
   - `RT 135`
   - `FD 154`
   - `LT 90`
   - `FD 15`
   - `RT 90`
   - `FD 10`
   - `RT 90`
   - `FD 63`
   - `RT 90`
   - `FD 10`
   - `RT 90`
   - `FD 16`
Use the commands below to help the turtle draw each design. Record the commands you use.

**FORWARD (FD)**
**BACK (BK)**
**RIGHT (RT)**
**LEFT (LT)**
**SHOWTURTLE (ST)**
**HIDETURTLE (HT)**
**PENDOWN (PD)**
**PENUP (PU)**
**HOME**

1. ![Square]
   
   **Commands:**

2. ![Triangle]
   
   **Commands:**

3. ![Zigzag]
   
   **Commands:**

4. ![Octagon]
   
   **Commands:**
Turtle Creations

You can make the turtle draw your initials, or a lightning bolt, or a face, or anything else you can imagine. Use the commands below to create your own design. Record the commands you use and sketch the design the turtle draws.

FORWARD (FD) PENDOWN (PD)
BACK (BK) PENUP (PU)
RIGHT (RT) DRAW
LEFT (LT) HOME

Commands: Sketch:
Lesson 3 - Procedures

Lesson Focus

To enter a LOGO procedure into the computer, discover the result of executing the procedure, and edit procedures.

Materials

LOGO on the classroom computers, copies of Activities 6 and 7, papers, pencils, and chalkboard.

Group Size

Whole-class, and individual (or partners).

Time for Activities

Each student on the computer for Activity-6 20 minutes and for Activity-7 30 minutes.

Teaching Notes

1. The students should have had experience with LOGO primitive commands DRAW, FORWARD, BACK, RIGHT and LEFT. They should be familiar with the general idea of procedure as a set of instructions for performing a task.

2. Distribute copies of Activity 6 to the students. Allow them to work independently.

3. For exercise 2.e the students should draw (roughly) the resulting square, illustrated below:

```
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>_____</td>
</tr>
</tbody>
</table>
```
4. Help students understand how to teach the turtle to remember a procedure. Distribute copies of Activity 6 to the students. Allow them to work independently.

5. Remind students that the computer will not check for mistakes. They must work carefully and check themselves. Remind students to use the correct spelling of the procedure name.

6. After students finish working with Activity 6, distribute copies of Activity 7. Tell the students that now they are going to learn how to change the turtle’s mind using the EDIT command. Then allow them work independently (or by partners) on the computers.
Name ___________________________  Activity 6

Teaching the Turtle to Remember

This activity presents the use of procedures to help the turtle remember.

A procedure is a step-by-step description of doing something.

1. Type the following words and press RETURN.
   a. SQUARE
   b. RECTANGLE

   Does the turtle know these words? ______

   What message do you see when you give each command? __________

2. Teach the turtle the procedure SQUARE by following the steps below.
   a. First enter the EDIT mode and tell the turtle the name of the procedure you want it to remember by typing:
      
      TO SQUARE
      
      Any other name can be used for this purpose, for example, BOX.
      
      b. Define the command in the procedure SQUARE by typing:
         
         FD 40  
         RT 90  
         FD 40  
         RT 90  
         FD 40  
         RT 90  
         FD 40  
         RT 90  

   c. Indicate that you are finished defining the procedure SQUARE by typing: END
   d. Press CONTROL and C at the same time to define the procedure SQUARE.
   d. Now the turtle knows how to SQUARE. Sketch:
      Type SQUARE and press RETURN. Then sketch what the turtle draws.

3. Teach the turtle the procedure TRIANGLE.

   TO TRIANGLE
   
   Sketch:
   FD 40
   RT 120
   FD 40
   RT 120
   FD 40
   RT 120
   END

   Press CONTROL and C at the same time to define the procedure TRIANGLE. Now type TRIANGLE, press RETURN and sketch what the turtle draws in the space provided above.
Activity 6 continued ...

4. Teach the turtle the procedure HOUSE. Notice that this procedure uses the procedures SQUARE and TRIANGLE as commands.

```
TO HOUSE
  SQUARE
  FD 40
  RT 30
  TRIANGLE
END
```

Press CONTROL and C at the same time to define the procedure HOUSE. Type DRAW and press RETURN. Now type HOUSE, press RETURN and sketch what the turtle draws in the space provided above.

5. See what new words are in the turtle's dictionary in the memory of the computer by typing:

```
PRINTOUTITLES OR POTS
```

Record the New Words the turtle knows.

The chart below shows ten useful editing commands. They are to be used in the EDIT mode of the computer. Remember that the computer will not check for mistakes. Observe that the notation CONTROL-D means to depress the CONTROL key and press the D key at the same time.

<table>
<thead>
<tr>
<th>KEY (S) TO PRESS</th>
<th>TO DO</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESC</td>
<td>back up and erase one character</td>
</tr>
<tr>
<td>----&gt;</td>
<td>go right without erasing</td>
</tr>
<tr>
<td>&lt;----</td>
<td>go left without erasing (Backward)</td>
</tr>
<tr>
<td>CONTROL-D</td>
<td>delete the character under the cursor (Delete)</td>
</tr>
<tr>
<td>CONTROL-K</td>
<td>delete text from cursor position to end</td>
</tr>
<tr>
<td>CONTROL-N</td>
<td>go to next line (Next)</td>
</tr>
<tr>
<td>CONTROL-P</td>
<td>go to previous line (Previous)</td>
</tr>
<tr>
<td>CONTROL-O</td>
<td>open a space to insert a line (Open)</td>
</tr>
<tr>
<td>CONTROL-C</td>
<td>exit editor and add procedure to dictionary (Complete)</td>
</tr>
<tr>
<td>CONTROL-G</td>
<td>exit editor without defining procedure (Gone)</td>
</tr>
</tbody>
</table>
Changing the Turtle's Mind

Before you begin this activity, be sure the turtle knows the words SQUARE, and TRIANGLE that were taught in Activity 6.

Using EDIT, you can change the procedures that you have taught the turtle. To edit the procedure SQUARE, type:

```
EDIT SQUARE
```

The commands in the procedure SQUARE will appear on the screen. Use the editing commands listed in your Personal Logo Dictionary to make changes to the commands in the procedure.

1. Change the following procedures to make the turtle draw a different SQUARE, a different TRIANGLE, and a different HOUSE. Remember to type CONTROL-C to enter the new procedure into the turtle's dictionary in the memory of the computer.

   a. EDIT SQUARE
   b. EDIT TRIANGLE
   c. EDIT HOUSE

```
FD 50 RT 90 FD 50 FD 50 RT 120 FD 50 RT 90 FD 50 RT 90 FD 50 RT 90 FD 50 FD 50 END
```

   d. Give the command HOUSE to the turtle. You may need to fix some of the procedures if you have made any errors in the changes.

2. Change the commands in the procedures SQUARE, TRIANGLE and HOUSE to make the turtle draw a bigger HOUSE that have sides equal to 60 turtle steps. Record your new commands for each procedure below.

```
EDIT SQUARE EDIT TRIANGLE EDIT HOUSE
```

3. Edit the procedures the turtle already knows or create your own procedures to make a design. Remember you need to name your procedure, enter the procedure in the computer memory, and demonstrate that the procedure works by telling the turtle to execute the procedure. You may need to fix the procedure if you have made any errors. Record your commands for the procedure and sketch your design on paper. You can use the design you created in Activity 5 for this exercise.

```
Commands: Sketch:
```
Lesson 4 - Modify Procedures and Repeat Command

Lesson Focus

To modify (edit) a LOGO procedure to make it accomplish a new but similar task, name procedures, develop a procedure using the Logo primitive command REPEAT, and use procedures within a procedure to make the turtle draw a design.

Materials

LOGO on the computers, Activities 8 and 9, paper, pencils, and chalkboard.

Group Size

Whole-class, and individuals (or partners).

Time for Activities

Each student on the computer for Activity-8 20 minutes, and for Activity-9 30 minutes.

Teaching Notes

1. Load the LOGO language before students start working on the computers.

2. Remind them to type CONTROL-C to enter the procedure in the computer’s memory and to type the commands correctly.

3. Help students notice that the REPEAT command has three necessary components are necessary. They are the word REPEAT, the number of times you want something to repeat (in this case 4 times), and the actual set of instructions, enclosed in brackets (in this case FD 40 RT 90). The first time the line is read, the turtle will go upward 40 steps and turn 90 degrees to the right. The next time it will travel 40 steps toward the right of the screen and make a 90 degree turn so that it is facing down. The third time around it will proceed 50 steps again, turn right (facing the center now), and make its last 90 degree turn, leaving it where it began. The program is now finished.
4. Distribute copies of Activity 8, and allow them to work independently (or by partners) on the computers. They might need to edit some commands if the procedures do not work. Remind them to check spelling, spacing, and any special signs.

5. After students finish the above activity, distribute copies of Activity 9, and allow them to work independently (or by partners) on the computers.

6. If the students finish before the class period has expired, ask them to create their own designs using the REPEAT command, demonstrate that the procedures work by telling the turtle to execute them, and to record their commands for each procedure and sketch their designs.
Easier Ways for You

The primitive command REPEAT can save you some typing. This command uses brackets: [ ]

Help the turtle draw by entering the following procedures in the computer's memory. Demonstrate that the procedures work by telling the turtle to execute them. Type DRAW and press RETURN after each procedure is executed. In the space under each procedure, sketch what the turtle draws.

1. TO SQUARE
   FD 50
   RT 90
   FD 50
   RT 90
   FD 50
   RT 90
   END
   Sketch:

2. TO NEW.SQUARE
   REPEAT 4[FD 50 RT 90]
   END
   Sketch:

3. TO LINES
   REPEAT 18[FD 40 BK 40 RT 20]
   END
   Sketch:

4. TO TRIANGLE
   REPEAT 3[FD 40 RT 120]
   END
   Sketch:
Predicting with REPEAT

Read each procedure and sketch what you think the procedure will tell the turtle to draw. Decide on a word to use as the name of each procedure. Then check your prediction by telling the turtle to execute the procedure. If your prediction was inaccurate, correct your sketch. Do not use the same name twice.

1. TO ____________
   REPEAT 4[FD 20 BK 20 RT 90]
   END
   Sketch:

2. TO ____________
   REPEAT 5[FD 30 RT 72]
   END
   Sketch:

3. TO ____________
   REPEAT 6[FD 30 RT 60]
   END
   Sketch:

4. TO ____________
   REPEAT 9[FD 30 RT 40]
   END
   Sketch:

TO ____________
REPEAT 12[FD 30 RT 30]
END
Sketch:
Lesson 5 - More Repeat Command

Lesson Focus

To develop a procedure using the LOGO primitive command REPEAT, and use the Total Turtle Trip Theorem to develop a procedure.

Materials

LOGO on the computers, Activities 10 and 11, pencils, and chalkboard.

Group Size

Whole-class, and individuals (or partners).

Time for Activities

Each student (or by partners) on the computer for Activity 10 20 minutes, and for Activity 11 30 minutes.

Teaching Notes

1. Load the LOGO language before students start working on the computers.

2. Distribute copies of Activity 10 to the students. Allow them to work independently (or by partners) on the computers. Help them understand that when they write procedures to have the turtle construct regular polygons the sides must be the same length, all of the angles through which the turtle turns have the same measure, they are closed figures since the turtle always ends up exactly where it starts, and the total degrees turned is 360 degrees.

3. Students should notice that as the number of sides increases, the amount the turtle turns at each corner decreases. From the table, they can see that the sum of the angles through which the turtle turns is 360 degrees.

4. Distribute copies of Activity 11 to the students. Allow them to work independently (or by partners) on the computers.
The Turtle's Trip

The turtle turns through angles to change directions. When the turtle draws a figure, its trip is measured by the sum of all the angles through which it turns.

What do you think the turtle draws when given these commands? (Sketch your prediction.) How many turns does the turtle make? What is the angle of the turtle's turn each time? How many total degrees does the turtle turn through (total trip)? (Hint: $360/3$ means $360 ÷ 3$.)

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Sketch</th>
<th>Number of Turns</th>
<th>Degree in Each Turn</th>
<th>Total Degrees</th>
</tr>
</thead>
<tbody>
<tr>
<td>TO TRIANGLE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REPEAT 3[FD 30 RT 360/3]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>END</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TO SQUARE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REPEAT 4[FD 40 RT 360/4]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>END</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TO PENTAGON</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REPEAT 5[FD 50 RT 360/5]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>END</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TO HEXAGON</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REPEAT 6[FD 50 RT 360/6]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>END</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Enter the above procedures in the computer's memory, and demonstrate they work by telling the turtle to execute them.

These procedures draw regular polygons (all sides are the same size). When you write procedures to have the turtle construct regular polygons, which conditions are necessary? Write Yes or No in the space provided below for each statement.

The sides of the polygon must be the same length. _____

The sum of the angles through which the turtle turns is 360 degrees. _____

All of the angles through which the turtle turns have the same measure. _____

The degrees in each turn is the same as 360 degrees divided by the number of turns the turtle makes to draw a polygon. _____
Many Sides

Create procedures that teach the turtle to draw the regular polygons described below. A regular polygon is a figure with all sides the same length and all angles the same number of degrees. Record your procedures.

1. An 8-sided regular polygon (octagon) in which each side is 10 turtle steps.
   TO EIGHT

2. A 10-sided regular polygon (decagon) in which each side is 10 turtle steps.
   TO TEN

3. A 12-sided regular polygon in which each side is 10 turtle steps.
   TO TWELVE

4. An 18-sided regular polygon in which each side is 10 turtle steps.
   TO EIGHTEEN

5. A 20-sided regular polygon in which each side is 10 turtle steps.
   TO TWENTY

6. A 36-sided regular polygon in which each side is 10 turtle steps.
   TO THIRTY.SIX

7. When you command the turtle to THIRTY.SIX, what does the figure look like?
Lesson Focus

To develop and debug (edit) LOGO procedures that involve variables.

Materials

LOGO on the classroom computers, copies of Activity 12A, and pencils and paper.

Group Size

Whole-class, and individual (or partners).

Time for Activities

Each student at the computer for Activity 12A 50 minutes.

Teaching Notes

1. Load the LOGO language before students start working on the computers.

2. Distribute copies of Activity 12A to the students. Allow them to work independently. Help students notice that the space before colon is important.

3. Help students enter the procedures in the computer's memory and to demonstrate how it works by telling the turtle to execute the procedure. They should understand that they can use any name they want for the procedure and the variable.

4. Ask them to put their papers in their folders when they finish.
The General Turtle

When you give the command PENUP, the turtle follows the command and picks up the pen. But when you give the command FORWARD, the turtle gives you a message indicating that FORWARD needs more input. The turtle needs to know "how far" FORWARD.

You can create procedures like FORWARD that require input by using variable names like L, or LENGTH, or SIDE. They are used to represent numbers. Different inputs or values for the variable will make the turtle do different things. The colon (:) is used before the variable to tell the turtle that the next letter or word is used as a variable.

Teach the turtle each of the procedures below, which use input. Test each procedure by giving the turtle three commands. Describe or draw what the turtle does in response to each command.

<table>
<thead>
<tr>
<th>PROCEDURE</th>
<th>TEST COMMANDS AND TURTLE'S RESPONSES</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. TO LINE :L</td>
<td>a. LINE</td>
</tr>
<tr>
<td>Fc :L</td>
<td></td>
</tr>
<tr>
<td>BK :L</td>
<td></td>
</tr>
<tr>
<td>END</td>
<td></td>
</tr>
<tr>
<td></td>
<td>b. LINE 20</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>c. LINE 30</td>
</tr>
<tr>
<td>2. TO SQUARE :LENGTH</td>
<td>a. SQUARE</td>
</tr>
<tr>
<td>REPEAT 4 [FD :LENGTH RT 360/4]</td>
<td></td>
</tr>
<tr>
<td>END</td>
<td></td>
</tr>
<tr>
<td></td>
<td>b. SQUARE 30</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>c. SQUARE 50</td>
</tr>
<tr>
<td>3. TO TRIANGLE :SIDE</td>
<td>a. TRIANGLE</td>
</tr>
<tr>
<td>REPEAT 3 [FD :SIDE RT 360/3]</td>
<td></td>
</tr>
<tr>
<td>END</td>
<td></td>
</tr>
<tr>
<td></td>
<td>b. TRIANGLE 40</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>c. TRIANGLE 70</td>
</tr>
</tbody>
</table>
Create a procedure for each of the following problems. Record your procedures. Use a variable and the REPEAT command for this activity.

1. Teach the turtle to draw an equilateral triangle (sides and angles are the same size) in which you can input the length of each side.
   
   Procedure: TO TRI :N

2. Teach the turtle to draw a box (sides are equal) in which you can input the length of each side.
   
   Procedure: TO BOX :N

3. Teach the turtle to draw a regular pentagon (5 equal sides) in which you can input the length of each side.
   
   Procedure: TO PENTAGON :N

4. Teach the turtle to draw a regular hexagon (6 equal sides) in which you can input the length of each side.
   
   Procedure: TO HEXAGON :N

5. Teach the turtle to draw a regular octagon (8 equal sides) in which you can input the length of each side.
   
   Procedure: TO OCTAGON :N
Lesson 7A - More Variables

Lesson Focus
To develop and debug (edit) LOGO procedures that involve variables.

Materials
LOGO on the classroom computers, copies of Activity 13A, and pencils.

Group Size
Whole-class, and individual (or partners).

Time for Activities
Each student (or partners) on the computer for group for Activity-13A 40 minutes, whole class discussion 10 minutes.

Teaching Notes
1. Load the LOGO language before the students start working on the computers.

2. Distribute copies of Activity-13A to the students. Allow them to work independently. Help students understand what each procedure will make the turtle draw. Remind them that $360/3$ is the same as $360 \div 3$, and that these procedures use the Turtle Total Trip Theorem.

3. Help students develop a general procedure (they might call it POLYGON) that commands the turtle to draw a regular polygon with $N$ sides and with length of each side 20 turtle steps. The students should develop a procedure similar to this one:

```
TO POLYGON :N
  REPEAT :N[F'D 20 RT 360/:N]
END
```

Emphasize that the variable $N$ represents the number of turtle steps the turtle makes to draw each side of the regular polygon.

4. Ask them to put their papers in their folders when they finish. Notice that some students may not be able to complete the challenge part in the amount of time
provided. Possible procedures for this part are the following:

TO POLYGON :N :L
REPEAT :N[FD :L RT 360/:N]
END

TO SPIN.POLY :N :T :A
REPEAT :T[POLYGON :N RT :A]
END
Last time you were writing procedures to make the turtle draw regular polygons of any size using variables. In this activity you are going to do something similar using variables.

What do you think the turtle will draw when given these commands? Sketch your guess. Decide a word to use as the title or name of each procedure. Then check your prediction by teaching the procedure to the turtle. If your prediction was inaccurate, correct your sketch.

(Note: $360/3$ means $360 \div 3$.)

1. TO REPEAT 3 [FD 20 RT $360/3$] END
   Sketch:

2. TO REPEAT 4 [FD 20 RT $360/4$] END
   Sketch:

3. TO REPEAT 5 [FD 20 RT $360/5$] END
   Sketch:

4. TO REPEAT 6 [FD 20 RT $360/6$] END
   Sketch:

5. TO REPEAT 8 [FD 20 RT $360/8$] END
   Sketch:

6. TO REPEAT 9 [FD 20 RT $360/9$] END
   Sketch:

7. Develop a procedure that command the turtle to draw a regular polygon with N sides and with length of each side 20 turtle steps. Record your procedure. Use a variable and the REPEAT command to develop this procedure.
   Procedure:
The Turtle Dories

POLY is a procedure that commands the turtle to draw a regular polygon with :N sides and with the length of each side 30 turtle steps.

```
TO POLY :N
    REPEAT :N [FD 30 RT 360/:N]
END
```

SPIN.POLY is a procedure that commands the turtle to spin an :N-sided polygon a specified number of times, :T.

```
TO SPIN.POLY :N :T
    REPEAT :T [POLY :N RT 360/:T]
END
```

The designs below were created using SPIN.POLY. Find the command that create each design (predict). To check your answers, teach the turtle the two procedures and then give the turtle your commands.

8. ![Design](image) 9. ![Design](image)

```
SPIN.POLY
```

10. ![Design](image) 11. ![Design](image)

```
SPIN.POLY
```

Challenge 11.

12. Teach the turtle to draw any regular polygon in which you can input both the number of sides and the length of each side.

Procedure:

```
13. Teach the turtle to spin a polygon in which you can input:
   The number of sides of the polygon.
   The length of the sides.
   The number of times to repeat the drawing of the polygon
   The angle to turn between the drawing of one polygon and the next.

Procedure:
```
Lesson 8A - Still More Variables

**Lesson Focus**

To develop and debug (edit) LOGO procedures that involve variables.

**Materials**

LOGO on the classroom computers, copies of Activities 14A and 15A, and pencils.

**Group Size**

Whole-class, and individual (or partners).

**Time for Activities**

Each student (or partners) on the computer for Activity 14A 15 minutes, and for Activity 15A 35 minutes.

**Teaching Notes**

1. Load the LOGO language before students start working on the computers.

2. Distribute copies of Activity 14A to the students. Allow them to work independently (or by partners) on the computers. Do the same for Activities 15A.

3. Help students understand what the letter N stands for in this procedure. (Number of sides.) What the letter L stands for in this procedure? (The number of turtle steps on each side of the polygon.) What does $360/\cdot N$ mean in this procedure? (360 degrees divided by the number of sides the polygon has, this is the number of degrees the turtle turns each time.)

4. At the end of the lesson ask students to put their paper in their folders.
Susan wrote a procedure to have the turtle construct different regular polygons. Predict what the turtle will construct when given the commands. Record the total number of turtle steps the turtle will take to construct the polygon. This is the perimeter of the polygon. Test your predictions by telling the turtle to execute the procedure.

TO POLYGON :N :L
REPEAT :N[FD :L RT 360/:N]
END

Command | Perimeter | Sketch
--- | --- | ---
1. POLYGON 5 10 |  |  
2. POLYGON 3 70 |  |  
3. POLYGON 8 10 |  |  
4. POLYGON 10 5 |  |  
5. POLYGON 12 5 |  |  
6. POLYGON 36 10 |  |  
7. POLYGON 360 1 |  |  
8. PUZZLER! How would you have the turtle construct a circle? What is the circumference of the circle?
Repeat and Increment

TO MAZE :SIZE
REPEAT 4 [FD :SIZE RT 90 MAKE "SIZE :SIZE + 5]
END

What do you think the turtle will draw if you taught it the procedure MAZE described above and then gave it the command MAZE 20? MAZE 30? Sketch your guess. Then check your prediction by defining the procedure and giving the command. If your prediction was inaccurate, correct your sketch.

1. MAZE 20
2. MAZE 30
   Sketch: Sketch:

3. What is the name of the variable in the procedure MAZE? _______

4. What does this variable represent? ________________

   In this procedure when we say MAKE "SIZE :SIZE + 5, the turtle add 5 to whatever number is in SIZE. So SIZE gets bigger and bigger. Then when the turtle goes FD :SIZE, the turtle goes 5 further than it did the last time.

5. Edit this procedure such that the commands are repeated 40 times instead of 4 times. Demonstrate that the new procedure works by telling the turtle to execute the procedure.

6. Teach the turtle to draw a new procedure in which you can input the size and the degrees of the turns. Note that this new procedure needs a name and two variables. It is similar to the procedure MAZE but with one more variable. Record your commands for this new procedure. Demonstrate the your procedure works by telling the turtle to execute the procedure.

TO ___________________________
Lesson 9A - Variables and Repetition

Lesson Focus

To work with LOGO procedures involving repetition and variables.

Materials

LOGO on the classroom computers, copies of Activities 16A and 17A, and pencils.

Group Size

Whole-class, and individual (or partners).

Time for Activities

Each student (or partners) on the computer for Activity 16A 20 minutes, and for Activity 17A 30.

Teaching Notes

1. Load the LOGO language before students start working on the computers.

2. Distribute copies of Activity 16A to the students. Allow them to work independently (or by partners) on the computers. Discuss their answers as a whole class.

3. Note that NEW.MAZE 90 is based on a square, NEW.MAZE 72 is based on a regular pentagon, NEW.MAZE 120 is based on a regular triangle, and NEW.MAZE 144 is based on a five point star. As they experiment with the NEW.MAZE procedure, students they will find many interesting patterns. Each of these patterns reflects the process of growth by which it was created. Ask students to analyze how the different patterns occur. For example, notice that the curvature of the maze arms is greater as you move farther away from 72. Remind them to type DRAW and press RETURN to clear the screen before drawing each NEW.MAZE.

4. Distribute copies of Activity 17A to the students. Allow them to work independently on the computers. Notice that this procedure uses three variables.

5. Ask students to put their papers in their folders.
Before you begin this activity, teach the turtle the following procedure. Demonstrate that the procedure works by telling the turtle to execute the procedure.

```
TO NEW.MAZE :ANGLE
MAKE "SIZE 0
REPEAT 100[FD :SIZE RT :ANGLE MAKE "SIZE SIZE + 2]
END
```

Then enter: WINDOW

The designs below were created using the NEW.MAZE procedure. Note that the value of the variable SIZE has been set up to start on zero and increase 2 each time (i.e. 0, 2, 4, 6, ..., 98). Remember to type CS and press RETURN before drawing each design on the screen. Note that each of the patterns reflects the process of growth by which it was created. Analyze how the different patterns occur.

1. [Image of design 1]
2. [Image of design 2]
3. [Image of design 3]
4. [Image of design 4]
5. [Image of design 5]
6. [Image of design 6]
SPIRAL is a procedure that commands the turtle to draw designs based on curves with different starting angles, angle increments, and sizes.

```
TO SPIRAL :SIZE :ANGLE :INCREMENT
REPEAT 720 [FD :SIZE RT :ANGLE MAKE "ANGLE :ANGLE + :INCREMENT]
END
```

The designs below were created using SPIRAL. Teach the turtle the procedure SPIRAL and then give the turtle your commands. Notice that there are three variables in this procedure.

1. SPIRAL 7 0 1

2. SPIRAL 7 0 7

3. SPIRAL 15 40 30

4. SPIRAL 20 1 20

You may wish to experiment some more with this procedure. It can create many beautiful pictures.

What will happen if the increment is zero? For example, SPIRAL 12 90 0
Lesson 10A - Create Procedures Using Variables

Lesson Focus

To develop and debug (edit) LOGO procedures that involve variables, repetition and increments.

Materials

LOGO on the classroom computers, copies of Activity 10A, and pencils.

Group Size

Whole-class, and individual (or partners).

Time for Activities

Each student on the computer for Activity 10A 50 minutes.

Teaching Notes

1. Load the LOGO language before the students start working on the computers.

2. Distribute copies of the Activity 10A to the students. Allow them to work independently. Ask them to put their papers in their folders when they finish. Notice that some students may not be able to develop procedures for all the designs presented in this activity. Remind students that they have been working with procedures that involve variables, repetition and increments. Tell them that they are going to try to develop their own procedures using variables, repetition and increments.
Create procedures to teach the turtle how to draw the following designs. Use variables and the REPEAT command to develop these procedures. Do as much as you can. Record your procedures in the space provided.

1. Procedure:

2. Procedure:

3. Procedure:

4. Procedure:
Lesson Focus

To introduce the concept of variable, and solve equations by finding a number replacement for variable that will make the equation true.

Materials

Pencils, chalkboard, Activities 13B and 14B.

Group Size

Whole-class, and individual (or partners).

Time for Activities

Whole class discussion 5 minutes; each student working on Activity 13B 20 minutes and on Activity 14B 20 minutes; whole class discussion 5 minutes.

Teaching Notes

1. Write the equation $9 + 7 = N$ on the chalkboard. Ask, "What is the sum?" Then write the equation again with the sum in the place of the $N$, $(9 + 7 = 16)$.

2. Distribute copies of activity 13B to the students. Allow them to work independently.

3. Distribute copies of Activity 14B to the students. Allow them to work independently. Discuss some of the exercises with the whole class.

4. Treat the variables that appear in these exercises in a natural way by pointing out that letters are sometimes used to reserve a place for a number.
Name ____________________________ Activity 13B

Variables

The letter N in each equation saves a place for the sum, difference, product or quotient. The letter N can represent any number, but there is only one number that makes the equation true or correct. Any other letter can be used for this purpose. Letters used in this way are called variables.

Copy each equation, replacing N with the sum, difference, product or quotient, for example,

a. 6 + 9 = N  
6 + 9 = 15  
b. 3 \times 6 = N  
3 \times 6 = 18

1. 6 + 4 = N  
2. 7 + 5 = N  
3. 9 + 4 = N

4. 10 - 6 = N  
5. 12 - 7 = N  
6. 13 - 4 = N

7. 2 \times 4 = N  
8. 9 \times 5 = N  
9. 8 \times 7 = N

10. 24 \div 3 = N  
11. 12 \div 2 = N  
12. 40 \div 8 = N

Copy each equation, replacing the variable with the sum, difference, product or quotient.

13. 11 + 9 = B  
14. 10 \times 8 = S  
15. 16 - 9 = X

16. 54 \div 6 = M  
17. 0 \div 4 = B  
18. 9 \times 3 = N

19. 8 \times 0 = S  
20. 55 - 0 = N  
21. 14 \times 1 = T

22. 54 + 0 = X  
23. 12 - 9 = Z  
24. 67 + 1 = N

25. 3 + 6 + 2 = B  
26. 5 + 4 + 5 = X  
27. (6 + 2) \times 3 = K

28. 7 + 2 + 5 = Z  
29. 12 + 5 + 6 = B  
30. 8 - (3 \times 2) = R
Find the Replacement

We solve an equation such as \( N \times 8 = 56 \) by finding a number replacement for \( N \) that will make the equation true.

What number times 8 is 56?

For example: \( N \times 8 = 56 \rightarrow 7 \times 8 = 56 \rightarrow N = 7 \)

The solution to the equation is 7.

Solve each equation by finding a number replacement for the variable.

1. \( N + 5 = 13 \) \( N = \) __ __

2. \( N - 8 = 9 \) \( N = \) __ __

3. \( N \times 9 = 45 \) \( N = \) __ __

4. \( 8 \times N = 56 \) \( N = \) __ __

5. \( N - 6 = 9 \) \( N = \) __ __

6. \( 72 \div N = 8 \) \( N = \) __ __

7. \( 6 + X = 14 \) \( N = \) __ __

8. \( 8 \times B = 32 \) \( B = \) __ __

9. \( 13 - M = 8 \) \( M = \) __ __

10. \( 24 \div Z = 4 \) \( Z = \) __ __

11. \( N + 8 = 17 \) \( N = \) __ __

12. \( 34 - N = 36 \) \( N = \) __ __

13. \( K \times 9 = 54 \) \( K = \) __ __

14. \( 12 \times B = 12 \) \( B = \) __ __

15. \( F - 6 = 9 \) \( F = \) __ __

16. \( C - 5 = 9 \) \( C = \) __ __

17. \( R + 5 = 5 \) \( R = \) __ __

18. \( 56 - H = 10 \) \( H = \) __ __
Lesson 7B - Variables and Expressions

Lesson Focus

To introduce the concept of variable as it is used in expressions, and to evaluate expressions for given replacements of the variable.

Materials

Pencils, chalkboard, Activities 15B, 16B and 17B.

Group Size

Whole-class, and individual (or partners).

Time for Activities

Whole class discussion 10 minutes; each student working on Activity 15B 20 minutes, on Activity 16B 20 minutes, and on Activity 17B if necessary for more practice.

Teaching Notes

1. Write \( N + 8 \) on the chalkboard and tell the students that we call this an expression. Show the students how to evaluate the following expressions if \( T \) is 10: \( 6T, 5 + T, T/2, T - 7 \). Then have the students choose another value for \( T \) and evaluate each expression.

2. Distribute copies of activity 15B to the students. Have them read the material at the top of the handout. Explain that although neither Dave's age nor Jean's is known, we know how their ages are related. We can use any letter, or variable, to represent Dave's age. When we know Dave's age, the expression \( D - 4 \) tells what we must do to find Jean's age. Show how to evaluate \( D - 4 \) for \( D = 10, 13 \), and 20.

3. Direct students' attention to the examples in the table. Point out that the phrase "5 times \( T \)" can be written as \( 5 \times T, 5 \ T, \) or \( 5T \), and that \( K \div 3 \) can be written as \( K/3 \). In algebra, the later form is used.

4. Use the warm up exercise as practice with the students to check their understanding of variable and evaluating expressions.

5. Distribute copies of Activity 16B to the
students. Allow them to work independently. As they work on exercises 1-27, check to see that they are substituting correctly in the expressions. In exercises 28-35, make certain students are writing the correct expressions. For example, in exercise 29 students should write $X - 8$, and not $8 - X$. Activity 17B is a practice supplement to be used only if necessary for extra practice.
Variables and Expressions

Jean is 4 years younger than her brother Dave. We can write an expression for Jean's age.

Let $D$ = Dave's age.

$D - 4$ = Jean's age.

The letter $D$ is a variable. Any letter could be used as the variable.

To evaluate an expression, we substitute a number for the variable and perform the operations.

<table>
<thead>
<tr>
<th>Dave's Age</th>
<th>Jean's Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>13</td>
<td>9</td>
</tr>
<tr>
<td>20</td>
<td>16</td>
</tr>
</tbody>
</table>

Study the examples in the table below.

<table>
<thead>
<tr>
<th>Expressions in words</th>
<th>Variable</th>
<th>Expressions with a variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 times a number $T$</td>
<td>$T$</td>
<td>$5T$ or $5 \times T$</td>
</tr>
<tr>
<td>A number $K$ divided by 3</td>
<td>$K$</td>
<td>$K/3$ or $K \div 3$</td>
</tr>
<tr>
<td>The sum of $B$ and 7</td>
<td>$B$</td>
<td>$B + 7$</td>
</tr>
<tr>
<td>6 minus a number $N$</td>
<td>$N$</td>
<td>$6 - N$</td>
</tr>
</tbody>
</table>

Warm Up

Name the variables in each expression:

1. $N + 7$  
2. $R - 6$  
3. $9Y$  
4. $T/7$  
5. $17Z$

Evaluate each expression:

6. $5T$  
   if $T$ is 7  
   if $K$ is 24

8. $A + 7$  
   if $A$ is 15  
   if $Z$ is 8
Evaluate each expression.

1. \( N + 11 \)  
   if \( N \) is 10, ____  

2. \( 8M \)  
   if \( M \) is 7, ____  

3. \( 100 - S \)  
   if \( S \) is 65, ____  

4. \( H/2 \)  
   if \( H \) is 40, ____  

5. \( 13B \)  
   if \( B \) is 4, ____  

Complete each sentence by evaluating the expressions.

6. If \( T \) is 7, then \( 9T \) is ____  

7. If \( C \) is 11, then \( 9C \) is ____  

8. If \( X \) is 19, then \( X - 12 \) is ____  

9. If \( F \) is 1, then \( F + 99 \) is ____  

10. If \( M \) is 20, then \( M/5 \) is ____  

11. If \( F \) is 87, then \( F/3 \) is ____  

12. If \( Y \) is 19, then \( 4Y \) is ____  

13. If \( J \) is 47, then \( 100 - J \) is ____  

14. If \( X \) is 8, then \( X/8 \) is ____  

15. If \( H \) is 9, then \( 16 - H \) is ____  

Complete each sentence by evaluating the expressions.

<table>
<thead>
<tr>
<th>( K )</th>
<th>( K + 12 )</th>
<th>( Z )</th>
<th>( 4Z )</th>
<th>( B )</th>
<th>( 17 - B )</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>3</td>
<td>20</td>
<td>7</td>
<td>24</td>
<td>10</td>
</tr>
<tr>
<td>17</td>
<td>9</td>
<td>21</td>
<td>10</td>
<td>25</td>
<td>9</td>
</tr>
<tr>
<td>18</td>
<td>15</td>
<td>22</td>
<td>15</td>
<td>26</td>
<td>6</td>
</tr>
<tr>
<td>19</td>
<td>38</td>
<td>23</td>
<td>25</td>
<td>27</td>
<td>1</td>
</tr>
</tbody>
</table>

Write an expression.

28. 4 times a number \( T \) ____  

29. 8 less than a number \( X \) ____  

30. a number \( Z \) divided by 5 ____  

31. a number \( K \) divided by 7 ____  

32. 20 decreased by a number \( Q \) ____  

33. 9 more than a number \( N \) ____  

34. the product of 8 and a number \( D \) ____  

35. the sum of a number \( W \) and 11 ____
Complete each table by evaluating the expressions.

1. \[ \begin{array}{c|c}
M & M + 8 \\
\hline
6 & 14 \\
9 & 17 \\
12 & 20 \\
22 & 28 \\
64 & 72 \\
\end{array} \]

2. \[ \begin{array}{c|c}
Y & Y - 6 \\
\hline
9 & 3 \\
15 & 9 \\
24 & 18 \\
32 & 26 \\
47 & 41 \\
\end{array} \]

3. \[ \begin{array}{c|c}
E & 3B \\
\hline
5 & 15 \\
10 & 30 \\
13 & 39 \\
21 & 63 \\
35 & 105 \\
\end{array} \]

4. \[ \begin{array}{c|c}
S & S - 2 \\
\hline
6 & 4 \\
18 & 16 \\
24 & 22 \\
34 & 32 \\
48 & 46 \\
\end{array} \]

5. \[ \begin{array}{c|c}
N & N + 12 \\
\hline
4 & 16 \\
10 & 22 \\
12 & 24 \\
31 & 43 \\
57 & 69 \\
\end{array} \]

6. \[ \begin{array}{c|c}
X & 10X \\
\hline
8 & 80 \\
15 & 150 \\
27 & 270 \\
59 & 590 \\
62 & 620 \\
\end{array} \]

Write an expression.

7. 9 times a number \( D \)

8. 7 more than a number \( T \)

9. 2 less than a number \( J \)

10. 75 decreased by a number \( L \)

11. a number \( E \) divided by 6

12. 8 times a number \( K \)

13. 12 minus a number \( W \)

14. a number \( T \) times 30

15. 18 more than a number \( C \)

16. a number \( F \) minus 15

17. 42 plus a number \( G \)

18. 39 divided by a number \( U \)

19. the sum of a number \( P \) and 9
Lesson BB - More Variables and Expressions

Lesson Focus

To evaluate expressions for given replacements of the variable.

Materials

Pencils, chalkboard, Activities 18B, 19B and 20B.

Group Size

Whole-class, and individual.

Time for Activities

Whole class discussion 5 minutes, each student working on Activity 18B 20 minutes, on Activity 19B 25 minutes, and on Activity 20B if necessary for more practice.

Teaching Notes

1. Review variables and expressions with the students. Remind them that any letter can be used as a variable, and that to evaluate an expression we substitute a number for the variable and perform the operations. The variable is used to represent a number in an expression or equation.

2. Distribute copies of Activity 18B to the students. Allow them to work independently. This activity reteaches the idea of variable and expressions and gives extra practice to the students in this area.

3. Distribute copies of Activity 19B to the students. Allow them to work independently.

4. Activity 20B is a practice supplement to be used only if necessary to provide enrichment experiences in this area.
Name ____________________________

Activity 18B

Reteaching Supplement Variables and Expressions

Expressions in Words | Expressions with a Variable | Substitute 6 for the Variable and Evaluate
---|---|---
The sum of 7 and Y | \( 7 + Y \) | \( 7 + Y = 7 + 6 + 13 \)

2 less than a number B | \( B - 2 \) | \( B - 2 = 6 - 2 = 4 \)
4 times a number Z | \( 4Z \) | \( 4Z = 4 \cdot 6 = 24 \)
12 divided by a number T | \( 12/T \) or \( 12 \cdot \frac{1}{T} \) | \( 12 \cdot \frac{6}{6} = 6 \)

Evaluate each expression

1. \( N - 7 \) \( N = 10 \) \( N = 10 - 7 = 3 \)
2. \( X/3 \) \( X = 9 \) \( \frac{9}{3} = 3 \)
3. \( 8T \) \( T = 6 \) \( 8 \cdot 6 = 48 \)
4. \( P + 12 \) \( P = 5 \) \( P = 5 + 12 = 17 \)
5. \( N/4 \) \( N = 16 \) \( \frac{16}{4} = 4 \)
6. \( 20 - R \) \( R = 3 \) \( 20 - 3 = 17 \)

7. \( 21/K \) \( K = 7 \) \( 21 \div 7 = 3 \)
8. \( X + 5 \) \( X = 6 \) \( X = 6 + 5 = 11 \)
9. \( 17 - W \) \( W = 15 \) \( 17 - 15 = 2 \)

Complete each table by evaluating the expressions.

<table>
<thead>
<tr>
<th>M</th>
<th>SM</th>
<th>X</th>
<th>X + 10</th>
<th>C</th>
<th>15 - C</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td></td>
<td>0</td>
<td></td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>3</td>
<td></td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>9</td>
<td></td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td></td>
<td>36</td>
<td></td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td></td>
<td>54</td>
<td></td>
<td>44</td>
<td></td>
</tr>
</tbody>
</table>

Write an expression for each.

13. 3 more than a number D
14. the product of 6 and a number X
15. a number M divided by 8
16. a number M increased by 3
17. 4 times a number P
18. 20 divided by a number R
More practice

Complete each table by evaluating the expressions.

1. \[ \begin{array}{c|c|c}
  P & P + B & \text{ } \\
  6 & 0 & \text{ } \\
  9 & 3 & \text{ } \\
  15 & 9 & \text{ } \\
  21 & 36 & \text{ } \\
  34 & 54 & \text{ } \\
\end{array} \]

2. \[ \begin{array}{c|c|c}
  T & T/3 & \text{ } \\
  0 & 3 & \text{ } \\
  3 & 9 & \text{ } \\
  8 & 12 & \text{ } \\
  10 & 44 & \text{ } \\
\end{array} \]

3. \[ \begin{array}{c|c|c}
  N & N/2 & \text{ } \\
  4 & 10 & \text{ } \\
  8 & 12 & \text{ } \\
  10 & 44 & \text{ } \\
\end{array} \]

4. \[ \begin{array}{c|c|c}
  M & 8M & \text{ } \\
  0 & 0 & \text{ } \\
  1 & 8 & \text{ } \\
  5 & 40 & \text{ } \\
  10 & 80 & \text{ } \\
  11 & 88 & \text{ } \\
\end{array} \]

5. \[ \begin{array}{c|c|c|c}
  N & N + 12 & \text{ } & \text{ } \\
  4 & 10 & \text{ } & \text{ } \\
  5 & 17 & \text{ } & \text{ } \\
  6 & 24 & \text{ } & \text{ } \\
  10 & 39 & \text{ } & \text{ } \\
\end{array} \]

6. \[ \begin{array}{c|c|c|c|c}
  X & 6X & \text{ } & \text{ } & \text{ } \\
  0 & 0 & \text{ } & \text{ } & \text{ } \\
  1 & 6 & \text{ } & \text{ } & \text{ } \\
  7 & 42 & \text{ } & \text{ } & \text{ } \\
  9 & 54 & \text{ } & \text{ } & \text{ } \\
\end{array} \]

Write an expression:

7. 9 less than a number \( P \)

8. 5 more than a number \( W \)

9. 9 less than a number \( K \)

10. 97 decreased by a number \( P \)

11. a number \( K \) multiplied by 10

12. 9 times a number \( N \)

Use the given values of \( K \) to evaluate the expressions and complete the table.

\[ \begin{array}{c|c|c|c|c}
  K & K \cdot 2 & (K \cdot 2) - 4 & (K \cdot 2) - 4 + 8 & \text{ } \\
  4 & & & & \text{ } \\
  3 & & & & \text{ } \\
  7 & & & & \text{ } \\
  6 & & & & \text{ } \\
  8 & & & & \text{ } \\
  0 & & & & \text{ } \\
\end{array} \]
Use the given information to complete each table.

1. The expression $25 \cdot T$ tells us the number of times we blink in a given period of time ($T =$ the number of minutes)

<table>
<thead>
<tr>
<th>Time</th>
<th>$T$</th>
<th>1 minute</th>
<th>5 minutes</th>
<th>1 hour</th>
<th>8 hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of blinks</td>
<td>$25 \cdot T$</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. The expression $(34 - A) \div 2$ tells us how many hours of sleep we need each night based on our age. ($A =$ age in years up to 12)

<table>
<thead>
<tr>
<th>Age</th>
<th>$A$</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hours of sleep</td>
<td>$(34 - A) \div 2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3. The expression $C/7 + 3$ tells us what the temperature is (in degrees Celsius) based on the number of chirps of a cricket. ($C =$ the number of chirps of a cricket in one minute)

<table>
<thead>
<tr>
<th>Chirps/minute</th>
<th>$C$</th>
<th>49</th>
<th>63</th>
<th>70</th>
<th>91</th>
<th>112</th>
<th>161</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature ($^\circ$C)</td>
<td>$C/7 + 3$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Lesson 9B - Math Machine

Lesson Focus

To provide practice regarding the concept of variable, and to evaluate expressions for given replacements of the variable.

Materials

Pencils, chalkboard, Activity 21B and 22B

Group Size

Whole-class, and individual.

Time for Activities

Whole class discussion 5 minutes each student working on Activity 21B 35 minutes; whole class discussion 10 minutes.

Teaching Notes

1. Review variables and expressions with students. A variable is used to represent possible values or numbers. Any letter can be used as a variable.

2. Distribute copies of Activity 21B to the students. Allow them to work independently. Discuss their findings as a whole class.

3. Do the same for activity 22B.
For each input number (N) the rule given by N + 8 tells the machine how to find the output. The output numbers for each of the input numbers 6, and 10 are given below using the rule N + 8. Complete the table by finding the output numbers of 16, 25 and 100 using the same rule, N + 8.

1. \[
\begin{array}{ccc}
\text{INPUT} & \text{N + 8} & \text{OUTPUT} \\
6 & 6 + 8 & 14 \\
10 & 10 + 8 & 18 \\
16 & \_ & \_ \\
25 & \_ & \_ \\
100 & \_ & \_ \\
\end{array}
\]

Find the output numbers for each of the input numbers using the rule inside the Math Machine.

2. \[
\begin{array}{c|c}
\text{INPUT} & \text{OUTPUT} \\
0 & \_ \\
1 & \_ \\
28 & \_ \\
67 & \_ \\
345 & \_ \\
\end{array}
\]
Activity 21B continued ....

3. INPUT | OUTPUT
---|---
0 | 
1 | 
2 | 
3 | 
5 | 
17 | 
20 | 

4. INPUT | OUTPUT
---|---
2 | 
12 | 
186 | 
32 | 
78 | 

RULE
\[ 3 \times R \]

RULE
\[ S \div 2 \]
Name ____________________________  Activity 22B

What's My Rule?

The input and output numbers of the following Math Machines are given below each machine. Find the rule for each Math Machine.

1. INPUT | OUTPUT
   0 | 2
   1 | 3
   2 | 4
   3 | 5
   4 | 6
   10 | 12

   What's the rule? _______________________

2. INPUT | OUTPUT
   0 | 0
   1 | 5
   2 | 10
   3 | 15
   4 | 20
   10 | 50

   What's the rule? _______________________
Activity 22B continued....

3. INPUT OUTPUT
   3  9
   4  1
   5  2
   6  3
   7  4
  10  7

What's the rule?

4. INPUT OUTPUT
   0  6
   1  7
   2  8
   3  9
   4 10
  10 16

What's my rule?
Lesson 10B - More Math Machines

Lesson Focus

To provide practice regarding the concept of variable, and evaluate expressions for given replacements of one and two variables.

Materials

Pencils, chalkboard, Activities 23B, 24B and 25B.

Group Size

Whole-class, and individual.

Time for Activities

Each student working on Activity 23B 20 minutes, Activity 24B 20 minutes and Activity 25B 10 minutes.

Teaching Notes

1. Remind students that they have been working with variables and expressions. A variable is used to represent possible values or numbers. Any letter can be used as a variable.

2. Distribute copies of Activity 23B to the students. Allow them to work independently. Do the same for activities 24B and 25B.
For each input number (N) the rule given by $2N + 1$ tells the machine how to find the output. The output numbers for each of the input numbers 2, and 12 are given below using the rule $2N + 1$. Complete the table by finding the output numbers of 16, 25 and 100 using the same rule, $2N + 1$. Note that $2N$ is the same as $2 \times N$.

1. | INPUT | $2N + 1$ | OUTPUT |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>$2(2)$ + 1</td>
<td>5</td>
</tr>
<tr>
<td>12</td>
<td>$2(12)$ + 1</td>
<td>25</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Find the output numbers for each of the input numbers using the rule inside the Math Machine.

2. | INPUT | OUTPUT |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td></td>
</tr>
<tr>
<td>67</td>
<td></td>
</tr>
<tr>
<td>345</td>
<td></td>
</tr>
</tbody>
</table>
Activity 23B continued ....

3. **INPUT**
   - 2
   - 4
   - 6
   - 8
   - 10
   - 18
   - 20

   **OUTPUT**

   **RULE**
   
   \( B \div 2 + 5 \)

4. **INPUT**
   - 2
   - 12
   - 6
   - 15
   - 8

   **OUTPUT**

   **RULE**
   
   \( 5n + 8 \)
What's My Rule?

The input and output numbers of the following Math Machines are given below each machine. Find the rule for each Math Machine.

**1.**

<table>
<thead>
<tr>
<th>INPUT</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>13</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>10</td>
<td>34</td>
</tr>
</tbody>
</table>

What's the rule? 

**2.**

<table>
<thead>
<tr>
<th>INPUT</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>15</td>
</tr>
<tr>
<td>10</td>
<td>39</td>
</tr>
</tbody>
</table>

What's the rule?
The pairs of numbers in the first column represent the values for the variables A and B. The final result is found by using the rule given for each table. Use the rules to complete the tables.

### 1.

**Rule:** \( A \times B + 2 \)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>6</td>
<td>26</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>14</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

### 2.

**Rule:** \( A \times B - 7 \)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
</tr>
</tbody>
</table>
Activity 25B continued....

3. **Rule: \( A - B + 3 \)**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4,2</td>
<td></td>
</tr>
<tr>
<td>15,3</td>
<td></td>
</tr>
<tr>
<td>6,2</td>
<td></td>
</tr>
<tr>
<td>16,4</td>
<td></td>
</tr>
<tr>
<td>28,7</td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX D

Test of Logical Thinking

Item 1

House Paint #1

A painter uses four cans of paint to paint six rooms. How many rooms can be painted with six cans of paint?

a. 7 rooms
b. 8 rooms
c. 9 rooms
d. 10 rooms
e. other

Reason

1. The number of rooms compared to the number of cans will always be in the ratio of 3 to 2.

2. With more cans of paint, the difference will be less.

3. The difference in the numbers will always be two.

4. With four cans of paint the difference was 2. With six six cans of paint the difference would be two more.

5. There is no way of predicting how much paint is needed.
Item 2  House Paint #2

How many cans of paint are needed to paint eleven rooms?

a. 5 1/2 cans  
b. 7 cans  
c. 7 1/3 cans  
d. 9 cans  
e. other

Reason

1. The number of cans of paint compared to the number of rooms will always be in the ratio of 2 to 3.
2. If there are five more rooms, then 3 more cans are needed.
3. The difference in the numbers will always be 2.
4. The number of cans will be half the number of rooms.
5. There is no way of predicting the amount of paint.
Suppose you wanted to do an experiment to find out if changing the height of a ramp changed the distance a ball rolled off the end. Which sets of apparatus would you use?

- The highest ramp should be tested against the shortest.
- All sets need to be tested against each other.
- As the height is increased, the weight must be decreased.
- The heights should be the same but weights should differ.
- The heights should differ but the weights should be the same.

Reason

**a.** I and IV  
**b.** II and IV  
**c.** I and III  
**d.** II and V  
**e.** all of them
Suppose you wanted to do an experiment to find out if changing the weight of the ball changed the distance it rolled off the end of a ramp. Which sets of apparatus would you use?

- a. I and IV
- b. II and IV
- c. I and III
- d. II and V
- e. all of them

Reasons

1. The heaviest ball should be compared to the lightest.
2. All sets need to be tested against each other.
3. As the weight is increased, the height should be decreased.
4. The weights should be different but the heights should be the same.
5. The weights should be the same but the heights should be different.
An American tourist is sharing a compartment on a Swiss train with six people. Three speak only English and three speak only French. What are the chances of speaking to someone who speaks English on the first try?

a. 1 out of 2  
b. 1 out of 3  
c. 1 out of 4  
d. 1 out of 6  
e. 4 out of 6

Reasons

1. Four selections are needed because the three French speakers could be chosen in a row.

2. There are six people from which one English speaking person must be chosen.

3. One English speaking person needs to be selected from a total of three.

4. One half of the people speak English.

5. In addition to an English speaking person, three French speaking people could be selected from a total of six.
Item 6  The Coins and Rings

Three gold coins, four silver coins, and five copper coins are placed in a sack. Four gold rings, two silver rings and three copper rings are placed in the same sack.

What are the chances of pulling out a gold object on the first try?

a. 1 out of 2  
b. 1 out of 3  
c. 1 out of 7  
d. 1 out of 21  
e. None of the above

Reason

1. One gold object has to be selected from objects made from gold, silver, and copper.

2. 1/4 of the coins and 1/8 of the rings are made from gold.

3. It does not matter whether a coin or a ring is picked. One gold object needs to be selected from a total of 7 gold objects.

4. One gold object must be selected from a total of twenty-one objects.

5. 7 of the 21 objects in the sack are made from gold.
Item 7

The Gumball Machine

A boy has a penny to use in one of two gumball machines. The first machine has 30 red and 50 yellow gumballs; the second has 20 red and 30 yellows. He likes only red balls.

His chance of getting a red is greatest in the second machine?

a. Yes
b. No

Reason

1. There are 30 red in the first machine and only 20 in the second.
2. There are 20 more yellows in the first machine and only 10 more yellows in the second.
3. There are 50 yellows in the first machine and only 30 in the second.
4. There is a greater proportion of reds in the second machine.
5. There are more gumballs in the first machine.
Item B  

The Spotted Dogs

Seven large dogs and 21 small dogs are shown in the picture. Some dogs are spotted and others are not spotted.

Are large dogs more likely to have spots than small dogs?

a. Yes  

b. No

Reason

1. Some small dogs have spots and some large dogs have spots.

2. Nine small dogs have spots and only three large dogs have spots.

3. 12 of the 28 dogs are spotted and 16 of the 28 dogs are not spotted.

4. \( \frac{3}{7} \) of the large dogs are spotted and \( \frac{9}{21} \) of the small dogs are spotted.

5. 12 of the small dogs have no spots and only 4 of the large dogs have no spots.
A restaurant allows a choice of three types of bread, three types of meat and three types of spread.

<table>
<thead>
<tr>
<th>Bread</th>
<th>Meat</th>
<th>Spread</th>
</tr>
</thead>
<tbody>
<tr>
<td>wheat (W)</td>
<td>ham (H)</td>
<td>ketchup (K)</td>
</tr>
<tr>
<td>rye (R)</td>
<td>chicken (C)</td>
<td>mayonnaise (M)</td>
</tr>
<tr>
<td>pumpernickle (P)</td>
<td>turkey (T)</td>
<td>butter (B)</td>
</tr>
</tbody>
</table>

Each sandwich must contain bread, meat and spread. How many types of sandwich can be prepared using only one type of bread, one type of meat and one type of spread?

List all of the possible types of sandwiches in the spaces provided on the Answer Sheet. More spaces are provided than you will need. Two examples of different sandwiches are provided for you. (WHK, RCM)
Item 10  The Car Race

In an automobile race there is a Dodge (D), a Chevy (C), a Ford (F) and a Mercury (M). An observer predicts that the order of finish will be DCFM. In the spaces provided on the Answer Sheet list all other possible orders in which the cars might finish.

More spaces are provided than you will need.
APPENDIX E

Robustness Semantic Differential

Read each set of adjective pairs used to describe six aspects of your learning environment. For each adjective place a "check" in one of the five blanks that is nearest to describing your feeling about the particular aspect. For example, the adjective pair of "happy" and "sad" could be marked as follows.

happy ______:______:______:______:______:sad

The mathematics teacher is

1. boring ______:______:______:______:______:interesting
2. fresh ______:______:______:______:______:stale
3. meaningless ______:______:______:______:______:meaningful
4. important ______:______:______:______:______:unimportant
5. usual ______:______:______:______:______:unusual
6. powerful ______:______:______:______:______:weak
7. passive ______:______:______:______:______:active
8. thrilling ______:______:______:______:______:quieting
9. uneventful ______:______:______:______:______:action-packed
10. challenging ______:______:______:______:______:dull

The computer is

11. boring ______:______:______:______:______:interesting
12. fresh ______:______:______:______:______:stale
13. meaningless ______:______:______:______:______:meaningful
14. important ______:______:______:______:______:unimportant
15. usual ______:______:______:______:______:unusual
16. powerful ______:______:______:______:______:weak
17. passive ______:______:______:______:______:active
18. thrilling ______:______:______:______:______:quieting
19. uneventful ______:______:______:______:______:action-packed
20. challenging ______:______:______:______:______:dull

The mathematics activities are

21. boring ______:______:______:______:______:interesting
22. fresh ______:______:______:______:______:stale
23. meaningless ______:______:______:______:______:meaningful
24. important ______:______:______:______:______:unimportant
25. usual ______:______:______:______:______:unusual
26. powerful ______:______:______:______:______:weak
27. passive ______:______:______:______:______:active
28. thrilling ______:______:______:______:______:quieting
29. uneventful ______:______:______:______:______:action-packed
30. challenging ______:______:______:______:______:dull

276
The mathematics learning activities with the computer are

31. boring ______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______:______
APPENDIX F

Understanding of the Concept of Variable Instrument

Decide whether the following statements are true always, sometimes or never. Put a circle around the right answer. If you put a circle around "sometimes" explain when this statement is true.

Practice Items:

1. \( P + 9 = 16 \)
   - always
   - never
   - sometimes, that is when ____________________________________

2. \( 9Z + 3 = 21 \)
   - always
   - never
   - sometimes, that is when ____________________________________

3. \( X + 6 = 3 + 6 \)
   - always
   - never
   - sometimes, that is when ____________________________________

4. \( 8 + P = 8 + 2 \)
   - always
   - never
   - sometimes, that is when ____________________________________

5. \( X + 6 = 8 + 5 \)
   - always
   - never
   - sometimes, that is when ____________________________________

6. \( R + 7 = 9 + 3 \)
   - always
   - never
   - sometimes, that is when ____________________________________

7. \( 6A + 3 = 9 \)
   - always
   - never
   - sometimes, that is when ____________________________________

8. \( 32 + 6 = 18 \)
   - always
   - never
   - sometimes, that is when ____________________________________
7. \(2X + 3X = 5X\)
   - always
   - never
   - sometimes, that is when

8. \(9N + 7N = 16N\)
   - always
   - never
   - sometimes, that is when

9. \(A + 2B = A + 4C\)
   - always
   - never
   - sometimes, that is when

10. \(X + 3Y = 6Z + X\)
    - always
    - never
    - sometimes, that is when

11. \(2N = N + 2\)
    - always
    - never
    - sometimes, that is when

12. \(R + 2 = 2R\)
    - always
    - never
    - sometimes, that is when

13. \(3 + S = 3S\)
    - always
    - never
    - sometimes, that is when

14. \(5A = 5 + A\)
    - always
    - never
    - sometimes, that is when

15. \(10 + A = B + 10\)
    - always
    - never
    - sometimes, that is when

16. \(W + 6 = Y + 6\)
    - always
    - never
    - sometimes, that is when
17. \( A + B = B + A \)
   
   always
   never
   sometimes, that is when

18. \( R + N = N + R \)
   
   always
   never
   sometimes, that is when

19. \( 2A = 2B \)
   
   always
   never
   sometimes, that is when

20. \( 5D = 5E \)
   
   always
   never
   sometimes, that is when

21. \( M + 2N = 2M + N \)
   
   always
   never
   sometimes, that is when

22. \( 2E + F = E + 2F \)
   
   always
   never
   sometimes, that is when

23. \( A + 2B = 2B + A \)
   
   always
   never
   sometimes, that is when

24. \( 5Y + Z = Z + 5Y \)
   
   always
   never
   sometimes, that is when

25. \( A + 2B + 2C = A + 2B + 4C \)
   
   always
   never
   sometimes, that is when

26. \( X + 3Y + 3Z = X + 3Y + 6Z \)
   
   always
   never
   sometimes, that is when
27. \( A + B = B + C \)
   - always
   - never
   sometimes, that is when __________________________

28. \( R + N = N + M \)
   - always
   - never
   sometimes, that is when __________________________

29. \( A + B + C = C + A + B \)
   - always
   - never
   sometimes, that is when __________________________

30. \( M + N + Q = Q + N + M \)
   - always
   - never
   sometimes, that is when __________________________

31. \( A + B + C = A + B + D \)
   - always
   - never
   sometimes, that is when __________________________

32. \( M + N + Q = M + P + Q \)
   - always
   - never
   sometimes, that is when __________________________
APPENDIX G

Data Analyses
### Table 1

**Analysis of Variance for CTBS Scores**

<table>
<thead>
<tr>
<th>Source of Variance</th>
<th>df</th>
<th>Sum of Squares</th>
<th>Mean Square</th>
<th>F</th>
<th>p</th>
<th>R²</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>2</td>
<td>89.17</td>
<td>44.59</td>
<td>.60</td>
<td>.55</td>
<td>.01</td>
</tr>
<tr>
<td>Error</td>
<td>86</td>
<td>6427.27</td>
<td>74.73</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Corrected Total</td>
<td>88</td>
<td>6516.44</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note.** CTBS = Comprehensive Test of Basic Skills.
Table 2

**Analysis of Variance for TOLT Scores**

Dependent variable: Pre-treatment scores for TOLT

<table>
<thead>
<tr>
<th>Source of Variance</th>
<th>df</th>
<th>Sum of Squares</th>
<th>Mean Square</th>
<th>F</th>
<th>p</th>
<th>R²</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>2</td>
<td>.96</td>
<td>.48</td>
<td>.37</td>
<td>.69</td>
<td>.009</td>
</tr>
<tr>
<td>Error</td>
<td>86</td>
<td>110.17</td>
<td>1.28</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Corrected Total</td>
<td>88</td>
<td>111.13</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note.** TOLT = Test of Logical Thinking.
Table 3

**Analysis of Variance for Posttest Scores**

<table>
<thead>
<tr>
<th>Source of Variance</th>
<th>df</th>
<th>Sum of Squares</th>
<th>Mean Square</th>
<th>F</th>
<th>p</th>
<th>R²</th>
</tr>
</thead>
<tbody>
<tr>
<td>Treatment</td>
<td>2</td>
<td>174.12</td>
<td>87.06</td>
<td>4.41</td>
<td>.0151*</td>
<td>.093</td>
</tr>
<tr>
<td>Error</td>
<td>86</td>
<td>1698.64</td>
<td>19.75</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Corrected Total</td>
<td>88</td>
<td>1872.76</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source of Variance</th>
<th>df</th>
<th>Group Mean Difference</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contrast</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computer vs. Textbook</td>
<td>1</td>
<td>2.012</td>
<td>.10</td>
</tr>
<tr>
<td>Computer vs. Control</td>
<td>1</td>
<td>3.464</td>
<td>.01**</td>
</tr>
<tr>
<td>Textbook vs. Control</td>
<td>1</td>
<td>1.462</td>
<td>.10</td>
</tr>
</tbody>
</table>

* p < .05.  ** p < .01.  *** p < .001

**Note.** Duncan’s multiple-range tests were performed to analyze the significance of the contrasts at different alpha levels.
Table 4

**Analysis of Variance for Retention Test Scores**

<table>
<thead>
<tr>
<th>Source of Variance</th>
<th>df</th>
<th>Sum of Squares</th>
<th>Mean Square</th>
<th>F</th>
<th>p</th>
<th>R²</th>
</tr>
</thead>
<tbody>
<tr>
<td>Treatment</td>
<td>2</td>
<td>438.17</td>
<td>219.08</td>
<td>10.28</td>
<td>.0001***</td>
<td>.193</td>
</tr>
<tr>
<td>Error</td>
<td>86</td>
<td>1833.58</td>
<td>21.32</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Corrected Total</td>
<td>88</td>
<td>2271.75</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source of Variance</th>
<th>df</th>
<th>Group Mean Difference</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contrast</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computer vs. Textbook</td>
<td>1</td>
<td>4.690</td>
<td>.01**</td>
</tr>
<tr>
<td>Computer vs. Control</td>
<td>1</td>
<td>5.864</td>
<td>.01**</td>
</tr>
<tr>
<td>Textbook vs. Control</td>
<td>1</td>
<td>0.174</td>
<td>.10</td>
</tr>
</tbody>
</table>

* p < .05. ** p < .01. *** p < .001

**Note.** Duncan's multiple-range tests were performed to analyze the significance of the contrasts at different alpha levels.
VITA

Enrique Ortiz was born in Raleigh, North Carolina on June 12, 1955. He was reared on Santurce, Puerto Rico and graduated from Central High School in 1973. He attended Interamerican University in Puerto Rico and received a Bachelor of Arts degree in Secondary Education (Mathematics) in May, 1976. He was awarded a Master’s degree in Schools Administration and Supervision from Phoenix University in July, 1981.

The author taught elementary and secondary school mathematics in Puerto Rico for four years. Following the completion of his Master’s degree in administration and supervision of schools, he accepted a job as Mathematics Specialist of the Chapter II Federal Program in Puerto Rico. The author has experience as Mathematics Supervisor and Research Evaluator. In 1982, the author enrolled as a full-time doctoral student in the Louisiana State University Department of Curriculum and Instruction in the area of Secondary Education (Mathematics), with a minor in Educational Research.

While he was a full-time student, the author received an assistanship with the Department of Curriculum and Instruction. He has taught the elementary school mathematics method course during the last two years.
The author is married to Diana Ortiz. She received a Bachelor of Science degree in Biology from the University of Puerto Rico in May, 1984. The author and his wife have a two and a half years old son, Enrique Gabriel Ortiz.
Candidate: Enrique Ortiz

Major Field: Secondary Education (Mathematics)

Title of Dissertation: A comparison of a computer programming approach to a textbook approach in teaching the mathematics concept "variable" to sixth graders.

Approved: S. Diane Miller
Major Professor and Chairman

Major Professor and Chairman

Dean of the Graduate School

EXAMINING COMMITTEE:

Date of Examination:

July 2, 1987