COMPUTER PROGRAMMING AND KINDERGARTEN CHILDREN
IN TWO LEARNING ENVIRONMENTS

by

Dorothy Ruth Clouston

B.A.(Ed.), Memorial University of Newfoundland, 1979

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ARTS

in

THE FACULTY OF GRADUATE STUDIES

(Education)

We accept this thesis as conforming
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

September, 1988

©Dorothy Ruth Clouston, 1988
In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Education (Math)

The University of British Columbia
1956 Main Mall
Vancouver, Canada
V6T 1Y3

Date October 12, 1988
Abstract

Research Supervisor: Dr. Marvin Westrom

This study examined the appropriateness of introducing computer programming to kindergarten children. Three issues were explored in the research:

1. the programming capabilities of kindergarten children using a single keystroke program
2. suitable teaching techniques and learning environments for introducing programming
3. the benefits of programming at the kindergarten level.

The subjects for the study were 40 kindergarten students from a suburban community in British Columbia, Canada. All students used the single keystroke program, DELTA DRAWING. Two teaching techniques were used— a structured method and a guided discovery method. Quantitative data were collected by administering five skills tests (skills relating to programming) as pretests and posttests to both groups. A programming posttest was also given. Qualitative data were obtained by recording detailed observation reports for each of the 22 lessons (11 for each group), conducting an interview with each child at the end of the study and distributing a parent questionnaire.

It can be concluded that it is appropriate to introduce computer programming to kindergarten students. The children in this study showed they are capable of programming. All students
mastered some programming commands to instruct the "turtle" to move on the screen. DELTA DRAWING was determined to be a suitable means to introduce programming to kindergarten children.

A combination of a structured teaching method and a guided discovery method is recommended for introducing a single keystroke program. It was observed that students in a guided discovery learning environment are more enthusiastic and motivated than students in a structured environment. Students need time to explore and make discoveries, but some structure is necessary to teach specific commands and procedures which may otherwise not be discovered. Social interaction should be encouraged while children use the computer, however most kindergarten children prefer to work on their own computer. There was no significant difference between the two groups on all but one of the five skills tests for both the pretests and the posttests. On the Programming Test the two groups did not perform significantly different.

It can also be concluded that learning to program promotes cognitive development in certain areas. On all but one of the five skills test both the Structured Group and the Guided Discovery Group scored significantly better on the posttest than on the pretest.

Lesson observation reports, student interviews and responses on parent questionnaires suggested that the computer experience was positive and rewarding for the kindergarten students.
CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. THE PROBLEM</td>
<td>1</td>
</tr>
<tr>
<td>Statement of the Problem</td>
<td>5</td>
</tr>
<tr>
<td>Purpose of the Study</td>
<td>5</td>
</tr>
<tr>
<td>Assumptions of the Study</td>
<td>6</td>
</tr>
<tr>
<td>Limitations of the Study</td>
<td>6</td>
</tr>
<tr>
<td>2. REVIEW OF THE LITERATURE</td>
<td>7</td>
</tr>
<tr>
<td>Young Children and Computers - Pros</td>
<td>7</td>
</tr>
<tr>
<td>Young Children and Computers - Cons</td>
<td>10</td>
</tr>
<tr>
<td>Advantages of Computers</td>
<td>12</td>
</tr>
<tr>
<td>Problems with Computers and Young Children</td>
<td>13</td>
</tr>
<tr>
<td>Young Children and Programming</td>
<td>14</td>
</tr>
<tr>
<td>Justifications for Teaching Programming</td>
<td>14</td>
</tr>
<tr>
<td>Intellectual Benefits of Programming</td>
<td>15</td>
</tr>
<tr>
<td>Social Benefits of Programming</td>
<td>20</td>
</tr>
<tr>
<td>Negative Findings Regarding Programming</td>
<td>21</td>
</tr>
<tr>
<td>Can Young Children Program?</td>
<td>22</td>
</tr>
<tr>
<td>Why Logo?</td>
<td>27</td>
</tr>
<tr>
<td>What is Logo?</td>
<td>32</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Introducing Logo to Young Children</td>
<td>39</td>
</tr>
<tr>
<td>Summary of the Review of the Literature</td>
<td>43</td>
</tr>
<tr>
<td>3. METHODOLOGY</td>
<td>46</td>
</tr>
<tr>
<td>Questions of the Study</td>
<td>47</td>
</tr>
<tr>
<td>Programming Capabilities of Kindergarten Children</td>
<td>47</td>
</tr>
<tr>
<td>Teaching Technique / Learning Environment</td>
<td>48</td>
</tr>
<tr>
<td>Benefits of Programming</td>
<td>48</td>
</tr>
<tr>
<td>Sample</td>
<td>49</td>
</tr>
<tr>
<td>Design</td>
<td>50</td>
</tr>
<tr>
<td>Intervention</td>
<td>50</td>
</tr>
<tr>
<td>Treatments</td>
<td>51</td>
</tr>
<tr>
<td>Instrumentation</td>
<td>71</td>
</tr>
<tr>
<td>Pretests and Posttests</td>
<td>71</td>
</tr>
<tr>
<td>Other Types of Data Collection</td>
<td>76</td>
</tr>
<tr>
<td>Observation Reports</td>
<td>76</td>
</tr>
<tr>
<td>Student Interview</td>
<td>77</td>
</tr>
<tr>
<td>Parent Questionnaire</td>
<td>77</td>
</tr>
<tr>
<td>Procedure</td>
<td>78</td>
</tr>
<tr>
<td>Data Analysis</td>
<td>80</td>
</tr>
<tr>
<td>4. RESULTS - QUANTITATIVE SECTION</td>
<td>82</td>
</tr>
<tr>
<td>Pretest / Posttest Results</td>
<td>82</td>
</tr>
<tr>
<td>Numeral Recognition Test Results</td>
<td>84</td>
</tr>
<tr>
<td>Upper Case Letter Recognition Test Results</td>
<td>86</td>
</tr>
<tr>
<td>Directional / Positional Skills Test Results</td>
<td>87</td>
</tr>
<tr>
<td>Design Concepts Test Results</td>
<td>87</td>
</tr>
<tr>
<td>Procedural Thinking Test Results</td>
<td>91</td>
</tr>
</tbody>
</table>
Programming Test Results .............................................. 93

5. RESULTS - QUALITATIVE SECTION ........................................ 96

   Observation Reports - Structured Group ............................. 96

   Summary of Observation Reports for the
   Structured Group .................................................. 110

   Observation Reports - Guided Discovery Group ................... 111

   Summary of Observation Reports for the
   Guided Discovery Group ........................................... 123

   Student Interview .................................................. 125

   Parent Questionnaire ............................................... 133

6. CONCLUSIONS, IMPLICATIONS AND
   RECOMMENDATIONS .................................................. 139

   Conclusions - Programming Capabilities of
   Kindergarten Children ............................................ 145

   Suitability of Keystroke Programs ................................. 144

   Type of Programming Experience .................................. 146

   Conclusions - Teaching Technique and Learning
   Environment ........................................................ 151

   Conclusions - Benefits of Programming .............................. 157

   Implications for the Classroom .................................... 161

   Recommendations for Future Research ............................. 164

REFERENCES ............................................................. 168

APPENDIX

   A. Procedural Thinking Tests ....................................... 177

   B. Student Record Sheet - Procedural Thinking Test ............ 180

   C. Programming Test ............................................... 181

   D. Definition of Major and Minor Problems for
Programming Test ........................................... 182
E. Student Record Sheet - Scale for Programming Test ... 183
F. Student Interview ........................................ 184
G. Parent Questionnaire ................................. 185
H. Lesson Plans ............................................ 189
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Summary of Statistical Significance for Five Skills Tests</td>
<td>83</td>
</tr>
<tr>
<td>4.2</td>
<td>Numeral Recognition Test Means and Standard Deviations</td>
<td>84</td>
</tr>
<tr>
<td>4.3</td>
<td>Summary of ANOVA for Numeral Recognition Test</td>
<td>85</td>
</tr>
<tr>
<td>4.4</td>
<td>Upper Case Letter Recognition Test Means and Standard Deviations</td>
<td>86</td>
</tr>
<tr>
<td>4.5</td>
<td>Summary of ANOVA for Upper Case Letter Recognition Test</td>
<td>87</td>
</tr>
<tr>
<td>4.6</td>
<td>Directional / Positional Skills Test Means and Standard Deviations</td>
<td>88</td>
</tr>
<tr>
<td>4.7</td>
<td>Summary of ANOVA for Directional / Positional Skills Test</td>
<td>89</td>
</tr>
<tr>
<td>4.8</td>
<td>Design Concepts Test Means and Standard Deviations</td>
<td>90</td>
</tr>
<tr>
<td>4.9</td>
<td>Summary of ANOVA for Design Concepts Test</td>
<td>91</td>
</tr>
<tr>
<td>4.10</td>
<td>Procedural Thinking Test Means and Standard Deviations</td>
<td>92</td>
</tr>
<tr>
<td>4.11</td>
<td>Summary of ANOVA for Procedural Thinking Test</td>
<td>92</td>
</tr>
<tr>
<td>4.12</td>
<td>Summary of $t$-test for Programming Test</td>
<td>94</td>
</tr>
<tr>
<td>4.13</td>
<td>Item Analysis for Programming Test</td>
<td>95</td>
</tr>
<tr>
<td>5.1</td>
<td>Summary of Results for Sticker Test</td>
<td>98</td>
</tr>
<tr>
<td>5.2</td>
<td>Summary of Results for Maze</td>
<td>100</td>
</tr>
<tr>
<td>5.3</td>
<td>Summary of Results for Letter Project</td>
<td>103</td>
</tr>
<tr>
<td>5.4</td>
<td>Summary of Results for Command Sheets</td>
<td>103</td>
</tr>
<tr>
<td>5.5</td>
<td>Summary of Results for Shape Transparencies</td>
<td>105</td>
</tr>
<tr>
<td>5.6</td>
<td>Summary of Results for Students with Partners on Shape Transparencies</td>
<td>105</td>
</tr>
<tr>
<td>5.7</td>
<td>Summary of Results for Shape Directions (1)</td>
<td>106</td>
</tr>
<tr>
<td>5.8</td>
<td>Summary of Results for Shape Directions (2)</td>
<td>108</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>Figure 5.1</td>
<td>Computer drawing by a kindergarten student of a shark</td>
<td>120</td>
</tr>
</tbody>
</table>
ACKNOWLEDGEMENT

I sincerely thank the members of my examining committee for the time and effort they have spent on my thesis. My thanks go to Dr. Marvin Westrom, the chairperson, for his guidance and support throughout the study. I am grateful to Dr. Douglas Owens, a member of my committee, for his encouragement and direction throughout my Master's program. I also very much appreciate the interest Dr. Hillel Goelman, a member of my committee, showed in my topic.

My thanks also go to Dr. Walter Boldt for his assistance with the statistical analysis.

I am grateful to the students and their parents who so willingly participated in the study and to the school administration for permitting me to do the research.

Finally, I am thankful to my husband, Ross, for his patience and encouragement throughout the program and his careful analysis of the thesis.
Chapter 1

THE PROBLEM

Children today are growing up in an era of high technology. Computers, and more specifically microcomputers, have played an integral role in this revolution. Microcomputers are now quite inexpensive. As a result of their affordability and transportability, there has been an influx of microcomputers into both homes and schools.

In education, computers are being used in all areas of the curriculum for a variety of applications ranging from word processing to problem solving to teaching physics. All age levels, from the kindergarten classroom to college students, use this tool. Clarke (1985-86) justifies their use in the educational system. She notes, "Computers are providing the possibility of amplifying human ability, both by enabling users to undertake tasks which were previously considered too difficult, too tedious, or too time consuming and by providing a tool for the development of cognitive skills. Microcomputers also have the potential to serve as tools in creating and maintaining an enthusiasm for learning through their capacity to make learning stimulating, relevant and intrinsically exciting" (p. 32).

Microcomputers are being used more frequently in early childhood programs. There are many questions which must be asked when discussing the role of the computer in this area. What role
does (or could) a microcomputer play in early childhood education? Should young children be exposed to computers? At what age level should they first be introduced? Spencer and Baskin (1983) question what children can and should learn about computers and what they should learn first. What computer skills, concepts and knowledge can be expected of young children, and at what levels should these be introduced? Can children in Piaget's preoperational stage of development learn to program? Should programming, and in particular Logo, be introduced using a single keystroke program? What teaching methods are most appropriate for teaching children to program? These are only a few of the pertinent questions which must be addressed when discussing the role of the microcomputer in early childhood education.

Many of these issues have not yet been addressed in current research. Ellingham (1982), with reference to this problem, suggests that we are at such an early stage in the area of computers in primary education that no one can yet give any clear answers. In addition, Brady and Hill (1984) question the generalizability, validity and reliability of those studies that have been done. They note, "When reviewing the current research relating to young children and computers, it becomes clear that there is much more rhetoric than solid evidence, that current studies have used very small groups, have maintained few standard research controls, and usually involve a university setting."
(p. 50). However, since increasingly more primary students are now using computers, more studies are likely to be conducted in this area.

Despite the lack of rigorous research in certain areas, computers are being used with young children for a multitude of purposes. The literature documents many of these activities which tend to fall into three categories—the computer as a tutor, the computer as a tool, and the computer as a tutee. Some educators suggest that the computer be used as a tutor to teach the children certain concepts, skills and knowledge or to reinforce skills previously taught (for example, Computer Assisted Instruction programs). Obrist (1983) believes that CAI "is undoubtedly one of the best ways of introducing the computer to the teacher and the children. For the teacher using a computer with children for the first time these types of programs are exceptionally easy to understand, require only a little help from the teacher and are enthusiastically received by children" (p. 27). Educators have also suggested using the computer as a tool, for example for word processing. The third type of computer activity is to use the computer as a tutee, that is, for programming (Bandelier, 1982; Clements, 1983-84; Papert, 1980; Shumway, 1983). In this situation the children have the control over the computer rather than vice versa. They have to teach the computer what to do by giving it explicit directions to do a particular task. This is programming. Proponents of this view would suggest that it is a creative use of the machine--one in which children can utilize natural curiosity and progress at
their own rate. Some researchers believe CAI-type programs involve passive learning and are used as electronic flash cards (Williams, 1984) whereas programming involves active and creative learning because it is more discovery and play oriented (Chandler, 1984). All three applications, as a tutor, as a tool and as a tutee, have already been implemented in early childhood programs. The question is, does the computer belong there?

"People rightly question whether young children are able to use a computer or whether computers are physically, psychologically or socially harmful" (Gillingham, 1983, p. 2).

This paper will focus on young children learning to use the computer as a tutee, that is, learning to program. In the literature two programming languages are commonly referred to for use with young children, BASIC and Logo. Logo is the most frequently mentioned language and tends to be viewed as a young child's programming language. It is said to be "friendly and accessible" (Clements, 1983-84) and is a language which places "the child in a setting where he or she controls both the learning environment and the technology" (Markuson et al., 1983, p. 48). The structure of the BASIC language is too complex for young children. Therefore, Logo has been chosen as the foundation for this study due to its popularity in the literature for use with elementary school-aged children. It has been accepted as a means of introducing young children to programming. Curriculum guides for computer studies even suggest
using Logo with kindergarten students. This study will assess whether a single keystroke program which incorporates Turtle graphics is a suitable programming language for young children.

Statement of the Problem

Is it appropriate to introduce computer programming to kindergarten children? Three areas of this problem were investigated in this study:

1. issues relating to the capabilities of kindergarten children to program
2. issues relating to the teaching techniques employed to introduce a single keystroke program and the learning environment which is created
3. issues relating to the benefits of learning to program

These issues were broken down into eighteen specific questions in Chapter 3, the methodology section.

Purpose of the Study

The purpose of the study was to examine the appropriateness of introducing computer programming to kindergarten children. In particular, the study examined:

1. the capabilities of kindergarten students to program.
2. two teaching techniques for introducing a single keystroke program.
3. the effect of a teaching technique on learning to program.
4. the effect of a teaching technique on learning specific skills after learning to program.
5. the effect of learning to program on the acquisition of specific skills.
6. the effect of learning to program on social interaction.

Assumptions of the Study

It was assumed that:
1. the students would be motivated to work on the computer.
2. students would program to the best of their abilities.
3. the students would represent a variety of developmental levels and abilities.
4. the teacher as researcher would result in a study with valid conclusions.

Limitations of the Study

The following is a description of the limitations of this study:
1. the sample was from one elementary school and was a convenience sample.
2. the researcher was the teacher of the students.
3. the length of the study was limited to one month.
Chapter 2

REVIEW OF THE LITERATURE

This review of the literature will focus on the suitability of young children using computers and specifically learning to program in Logo. It includes a discussion of the benefits of young children learning to program, reasons opposing young children learning to program and suggested techniques for teaching Logo.

Young Children and Computers - Pros

"Paul Karoff presents a fairly comprehensive and unbiased account of the various positions concerning computers and the three to six-year-old age group. He states that 'the bottom line, of course, is that it is still too early to tell...'' (Ball, 1985, p. 374). Are computers appropriate for young children? If computers are to be used with these children there should be logical and legitimate reasons for doing so. Furthermore, within the school system, if microcomputers are to be used with preschool and primary school children, there should be a valid educational reason for doing so. One justification is to prepare these children for their adult life so that they may function in and contribute to a society in which computers play an integral role (Fetterman, 1981; Wayth, 1983; Jones, 1984). Fetterman (1981) adds that, "ignorance of technology can only
make people apprehensive, fearful and vulnerable." (p. 27). Many adults suffer from computerphobia (Doorly, 1980). To avoid this problem with the next generation, it is believed we should introduce the children to computers as early as possible. In addition, children not provided with computer experiences will possibly lag behind those that have had some exposure. Williams and Williams (1984) believe that if we do not provide computer training for students at an early age then we may have to compensate for the loss when they are older.

Educational equity is also cited as a key reason for beginning computer literacy in kindergarten so that all students may benefit from experiences with the computer (Hunter, 1983). If computer literacy is postponed to the higher grades, where specialization in subjects and streaming often occurs, then many students will not have the opportunity to work with the computer. Instead, only a select few will be given this privilege.

Another reason presented in favour of using microcomputers with young children through programming is that it makes the abstract more concrete (for example, through using Logo). Relationships that young children were previously not exposed to because of their complex nature are inherent features of Logo. Papert contends Logo makes these abstractions more understandable and concrete. "In his book, Mindstorms, Papert describes how Logo enables children to enter 'mathland', a place where they can explore sophisticated, advanced mathematical concepts such as differential geometry, but in terms and ways that they understand and enjoy" (Coburn et al., 1985, p. 67). If this is indeed so,
then children could possibly move through Piaget's stages of cognitive development at a faster rate (Papert, 1980; Piestrup, 1984; Willis et al., 1983). Some researchers also believe that using the computer to program promotes problem solving and logical or procedural thinking skills (Alper, 1980-81; Papert, 1980). Their reasoning is that in order for the children to direct the computer, they must determine a specific plan for carrying out the task. To do this, the children need to discover how they might carry out the plan and consequently must reflect on their own thinking (Papert, 1980).

A major criticism of using computers with young children is that computers have limitations and force children to perceive reality in a particular way. All playthings presented to children have their limitations and so computers are no different in this respect from other media (Silvern and McCary, date unknown). For example, one limitation of the computer that is often cited is that it limits the physical activity of the child. This is no different from other valuable early childhood activities such as reading, writing, drawing and painting (Silvern and McCary, date unknown) which also limit physical activity but would not be eliminated from an early childhood program because of this limitation. The computer can provide another valuable learning experience for the young child.
Young Children and Computers - Cons

Contrary to these beliefs are those who oppose the view that computers should play a role in early childhood education. For instance, Barnes and Hill (1983) believe children should not use the computer until they approach the concrete operational stage (approximately seven years old). They argue that children in the preoperational stage who use the computer are being deprived of concrete, three-dimensional experiences. They comment that, "Young children sitting at microcomputers are being short-changed in terms of fulfilling those needs which are fundamental to learning and to their optimal development...They are being limited to dealing with two-dimensional abstractions of real objects that are represented on a screen (p. 13)." Davy (1984) supports this view and describes the Logo environment as "perceptually impoverished" (p. 550). He contends that young children do not need to demonstrate computer competency. Instead, this can be left for the high schools. Obrist (1985) offers a caveat on the subject. "In introducing children to microcomputers at an early age we must be careful not to risk incurring that 'squeeze on infancy' that has deprived them of the right to go at their own pace" (p. 52). This opposition group also contends that young children in the preoperational stage of development do not possess the level of abstract thinking that is required to work on the computer (Barnes & Hill, 1983). At this stage children are believed to be inflexible in their thinking and unable to conserve. For a child to be able to successfully
debug a program he or she must be able to think flexibly. Otherwise, the child starts the problem again discarding in total the previous solution.

Still other opponents suggest that computers do not promote human interaction and communication and in fact may have a deleterious effect on social development (Larter, 1983; Barnes and Hill, 1983). They fear that the computer discourages social interaction. Barnes and Hill (1983) indicate that working on the microcomputer may be an isolating experience and may not promote communication skills. Studies have tended, however, to show the opposite is true. In a study conducted by Clements and Nastasi (1985) they found "that computers facilitate on-task cooperative work" (p. 6). These results were unexpected by the researchers. They were both surprised and encouraged that social interactions, similar to what occurs in play-oriented learning centers, took place when they worked on computers despite the abstract and cognitive aspect of the programs. These interactions were greater than in traditional school situations.

Yet another view is that young children do not possess the necessary reading, number and coordination skills to successfully use the computer (Larter, 1983). This problem has been overcome by using programs which limit the number of keys the child needs to use as well as limiting or eliminating the amount of written information presented on the screen.

Another perspective on this issue is that the age of introduction to a computer is a moot point. Proponents of this view say that the technology has arrived, so it will be used with
children of all ages and it is unlikely that this process will be reversed. (Spencer & Baskin; 1983, Reimer, 1985). In fact Larter (1983) states that, "The idea that there is a minimum age below which children should not be using microcomputers has been of almost no interest to the authors of articles... about the educational uses of microcomputers." (p. 62).

Advantages of Computers

If the movement to use microcomputers with young children has already begun, then the advantages to using these machines for instructional purposes should be identified. Many of the advantages which will be listed have not been supported to date by studies but instead are opinions based on observation or what "experts" in the field suggest. The benefits for using microcomputers with children include the following:

1. presenting material in various modes
2. encouraging cooperative work
3. increasing concentration span
4. providing immediate feedback
5. providing individualized instruction
6. providing motivation
7. increasing self-esteem
8. increasing confidence
9. developing thinking skills
10. developing eye-hand coordination
11. patience of the machine
12. nonthreatening and nonjudgemental environment
13. developing cognitive skills

Problems with Computers and Young Children

Even though computers are used freely with young children, the computer world for young children has its problems. One of the most fundamental difficulties in this area is the lack of good quality software (Gill, 1983; Williams, 1984). Educators and software developers need to coordinate their efforts to produce both educationally and technically sound programs. Another important problem is that many early childhood educators do not have sufficient training and are reluctant to use the computer in their classes (Williams & Williams, 1984). Teachers who do use computers tend to use CAI programs initially with their students rather than teaching a programming language such as Logo since CAI programs are easier and quicker to learn. Furthermore, there are very few developed computer literacy (or awareness) curriculums for the early childhood age range. Consequently, teachers must try to develop their own programs. These problems can be solved by time and involvement of larger numbers of people who include teachers, curriculum supervisors, developers of programming languages and producers of software packages.
Young Children and Programming

"The child as programmer. This is perhaps the most controversial aspect of the use of the computer in the Primary School" (Kent, 1984, p. 83). Many people are in favour of teaching young children programming (Bandelier, 1982; Birch, 1984; Clements, 1983-84; Hines, 1983; Jaworski and Brummel, 1984; Papert, 1980; Shumway, 1983) because they believe programming involves active and creative learning which is discovery and play oriented (Chandler, 1984) whereas CAI-type programs involve passive learning and are often used as electronic flash cards (Williams, 1984). There are, however, many opponents to young children learning to program (Barnes and Hill, 1983; Burg, 1984). They contend that computers, and more specifically programming, is too abstract for young children. This section will address both sides of this issue.

Justifications for Teaching Programming

Why should young children learn a programming language? "At present there seem to be three major justifications for the emphasis on computer programming in grades kindergarten to 12. "These are: 1) to prepare students for jobs and for post-secondary education; 2) to prepare students for citizenship in a computer-based society; and, 3) to use computer programming to enhance a student's general intellectual abilities" (Coburn et al., 1985, p. 63). These are very general reasons for
introducing computer programming to all students. There are more specific reasons stated for primary children. Bitter and Camuse (1984) state that learning Logo at an early age provides a positive experience and a foundation for later learning of more complex programming languages. Reimer (1985) reasons that it is important for young children to receive early exposure to computer programming because it will help prevent computerphobia as an older child and as an adult.

**Intellectual Benefits of Programming**

Besides preparing students for future programming in other computer languages and providing positive exposure to computers, does learning to program benefit the child cognitively or intellectually, or in any other ways? The research presents very conflicting results in this area even though much of the literature claims that Logo promotes cognitive and perceptual growth (Clements, 1984; Gillingham, 1983; Hines, 1983; Papert, 1980).

In a study conducted by Clements ("Cognition, Metacognition Skills and Achievement," 1985) with grade one and grade three children, the results indicated growth in several areas, which seemed attributable to Logo use. Seventy-two six-year-olds and eight-year-olds were randomly assigned to one of three treatment conditions: computer programming in Logo, computer-assisted instruction and a control group. Clements concluded that performance on certain cognitive and metacognitive skills, on
measures of creativity and giving directions can increase from learning Logo programming. Achievement in reading and mathematics was not significantly different for the three groups.

Papert (1980) claims that Logo promotes metacognitive skills because in order for the child to teach the computer how to perform a particular task, he must first examine how he thinks himself. "Thinking about thinking turns the child into an epistemologist" (p. 19). When children write or debug a program, they must go through this thought or problem solving process. Papert (1980) suggests that every program written in Logo develops problem solving strategies. His claims are supported by anecdotal evidence. Kieren (1984) suggests that, "...in all of its uses Logo is seen explicitly or implicitly as a problem solving task" (p. 28).

In a study by Hines (1983) at North Texas State University, six five-year-olds were given Logo instruction over a ten week period. The researcher aimed to determine whether kindergarten children could perform simple computer programming. Her conclusions were based on results of five children (one child moved). All of the children were capable of working out a computer program during the study. The conclusions stated that, "Five-year-old children are able to use the computer as a tool for thinking--for solving problems and learning about mathematics in a meaningful way" (p. 12).
Another study conducted by Reimer (1985) also explored the use of Logo with kindergarten children. The pilot study investigated the effect of learning to program in Logo on the child's readiness for grade one, on creativity and on self-concept. The study involved a treatment group and a control group with eight kindergarten students in each and it was a pretest-posttest design. The treatment group who had Logo computer experiences achieved more on the readiness, creativity and self-concept tests. However, the treatment group made statistically significant gains only on four of the eleven readiness tests. The gains made by this group on the creativity and self-concept tests were not statistically significant. Nevertheless, "results of this study suggest that age five kindergarten children can show greater gains in several commonly taught curriculum and developmental areas when they are involved in a Logo programming experience than children of the same age and grade level who do not have this experience" (p. 12).

Papert (1980) also contends that using Logo can permit young children to concretize formal operations well before the expected eleven or twelve years of age. The results of few studies have strongly supported this contention. Clements ("Cognition, Metacognition and Achievement Skills," 1985) reports that a few studies have shown weak evidence that Logo programming can increase performance on Piagetian operational tasks (Brown and Rood, cited in Clements, "Cognition, Metacognition and Achievement Skills," 1985; Hines, 1983). It is not known whether the age level of the child affects the results. It may only
occur in children who are making the transition from the 
preoperational to the concrete operational level (Clements, 

Many reports (Martin and Berner, 1982; Papert, 1972; Statz, 
cited in Howe et al., 1979) suggest that Logo can improve 
mathematical understanding because Logo is a mathematical and a 
geometrical environment. Kelman et al. (1983) discuss the impact 
of Logo on mathematics. They contend that the graphics computer 
can help overcome student's difficulty with visualizing geometric 
and trigonometric relationships and thus aid in their teaching. 
This will mean more emphasis will be put on "figure" in 
mathematics education whereas it presently emphasizes "number". 
Although Logo may be a mathematical environment, Leron (1985) 
warns educators that engaging in mathematical activities does not 
ensure that the children will learn the mathematical concepts 
involved. The learning process will have to assist the students 
in making the transition from using the program to formulating 
these concepts.

Several studies have investigated the effect of programming 
in Logo on mathematical achievement and understanding. Clarke 
(1985-86) conducted a pilot study with 43 girls in grades one, 
three and five in a girl's school. The students received two 20 
minute Logo computing sessions per week during the school year. 
Testing at the end of the year showed an increase on SAI scores 
and attitudes toward mathematics "suggesting the Logo experience 
had a positive effect on general ability and on expressed 
interest in learning about mathematics" (p. 33).
The focus of the Edinburgh Logo project by Howe et al. (1979) was to study the effect programming in Logo had on the mathematical achievement of children who experienced difficulty with mathematics. Eleven boys, all 13-years-old, were involved in a two year study. The experimental group's classroom mathematics performance improved more than the control group and their attitude was marginally more positive than the control group. The researcher concluded that the mathematical understanding of less able children can improve after using a mathematics program based on Logo. He also concluded that Logo can aid in learning geometry."

The findings of the MIT Brookline Project (1979), which observed children using Logo, were that the students' abilities to estimate length and angle developed considerably during the project after having Logo experiences (Papert, 1979).

In the Lamplighter Project, Logo was introduced to three- to nine-year-old children and the students were observed over four years. The conclusions were that the children gained a better understanding of mathematics concepts and enjoyed mathematical activities even though their performance on standardized tests did not necessarily improve (Watt, 1982). This study implies that Logo may not directly improve mathematical performance but it is a good motivator for the subject which in itself is a worthwhile use.
Reiber (1985) conducted a study with second grade children using Logo for a three month period. The results showed that the experience significantly increased the students' recognition of geometric shapes by the end of the study.

Social Benefits of Programming

Many educators fear that the computer is an isolating tool. They believe that the computer discourages social interaction. There have been studies, however, which have shown the opposite is true. In addition to intellectual benefits, studies have also suggested that learning to program has a positive social effect (Kieren, 1984). In a study conducted by Clements and Nastasi (1985) they found, as did previous research, "that computers facilitate on-task cooperative work" (p. 6). Clements ("Logo and CAI on Social Behaviours," 1985) reports that children actually prefer to work with others when using the computer than to work alone and consequently social and emotional development is fostered rather than inhibited.

Several studies and discussion papers in the literature have focused on the social and emotional development benefits of a Logo setting. "Tim Riordon claimed that the sharing of ideas is a fundamental part of the Logo environment" (Jaworski and Brummel, 1983, p. 23). Children programming in Logo tend to use their peers for help and talk to each other about their work more than during noncomputer classroom activities (Hawkins, 1983). In a study conducted by Silvern et al. (1988), 39 students ranging
from three-year-olds to seven-year-olds had access to a computer in a learning center. The study tentatively concluded that using the computer "supports, (and) perhaps even requires, social interaction" (p. 33).

In addition to social benefits others believe Logo can promote self-confidence. Reimer (1985) cites Berry (1981) as claiming that students have a higher self-esteem if they had a positive experience learning to program in Logo. Considerably more research on the social effects of using Logo needs to be done before definite conclusions can be drawn.

Negative Findings Regarding Programming

Not all of the research supports Papert's claims that Logo benefits a child intellectually and socially. "There has always been a strong demand especially from the outside, for the Logo community to supply research data to prove its grand claims regarding the educational value of Logo. Several 'evaluation studies' have recently appeared with negative findings" (Leron, 1985, p. 32).

For example, Statz (cited in Howe et al., 1979) found limited improvement in problem-solving skills as a result of learning to program. Pea, one of the researchers at the Bank Street School, has taken a closer look at the effects of learning Logo on procedural thinking and the results were disappointing. The structured thinking displayed during programming did not transfer to non-computer tasks (Burns, 1983; Tashner, 1984).
Students were using and producing the language without understanding it. These conclusions concur with the results of studies done with college computer science majors who had programmed for numerous hours but failed to understand basic principles that underlay even short programs (Tashner, 1984). "These studies, Pea concludes, do raise serious doubts about the sweeping claims made for the cognitive benefits of learning to program" (Tashner, 1984, p. 30). They imply that Logo is not the educational panacea Papert claimed it to be. Some children are experiencing difficulties with it and also they are not making the educational gains which were anticipated.

Can Young Children Program?

Can young children program? There are strong supporters on both sides of the issue. Those who oppose young children programming contend that computers and more specifically programming is too abstract for young children. They suggest that teachers and parents are expecting too much from the children, since the children have not developed the adequate logic skills to be able to understand programming (Barnes and Hill, 1983; Burg, 1984). Burg (1984) warns readers, "There is the danger that some will expect young children to be able to develop simple programs. Early childhood professionals must be ready to remind parents and others that a five-to-seven-year-
old's intellectual development is not sufficient for programming" (p. 32). These opponents use Piaget's cognitive stages of development to justify their argument.

There are four stages in Piaget's theory of intellectual development which include: the sensorimotor stage (birth to two years), the preoperational stage (two to seven years), the concrete operational stage (seven to eleven years) and the formal operational stage (eleven to fifteen years). The latter three stages are of concern in this discussion. The preoperational child is able to use representational thought but his or her thought is dominated by perceptions and little attention is paid to transformations. The concrete operational child is characterized by more freedom and control in thinking such as reversibility of thought. The child can think through an act but can also undo the act mentally without relying on physical actions. The formal operational child has flexibility, control, mobility and freedom of thought and can deal with abstractions (Smart and Smart, 1977).

Opponents of young children using the computer state that programming requires formal logical reasoning (Favaro, 1983) which does not develop until the formal operational stage of development. The concrete operational child's thinking is not usually logical and relies more on trial and error (Burg, 1984). Burg goes on to say that the young children's information processing abilities do not permit them to be able to plan a program. Children at this level "place one event after another with no attention to natural sequencing by cause and effect."
They cannot consider more than one aspect of a situation simultaneously. Consequently, they can neither imagine nor string together a set of commands to tell a computer what to do, even in the most simplest [sic] case" (p. 32). In addition, according to Piaget's theory, children under about seven years of age are basically state-oriented which means that they cannot relate states (orientation and location) and transformations (commands to alter state) (Cuneo, 1985). In order to debug a program, this flexibility of thought, which is not present in the young child, is necessary so the student can determine the error without starting again (Munro-Mavrias, 1983). In a pilot study conducted by Munro-Mavrias with grade one children, over a three month period, the ability to reverse thought processes as well as spatial-motor ability was related to success in programming a modified version of Logo.

Educators assume that because children can use Logo commands almost immediately then they have an understanding of their function. Cuneo (1985) conducted a study with four- to seven-year-olds to investigate the students' level of understanding of single turtle commands as transformations that connect turtle states. Logo is a language that is a system of states and transformations in which the states are defined by orientation and location and the transformations are the commands which result in a change of state. For the study, 32 four- and five-year-olds and 32 six- and seven-year-olds used a single keystroke Logo program. The researchers conclusions were not positive. The majority of the children misunderstood some or all
of the basic turtle commands. "The results suggest that young children's entry into turtle graphics programming will not be as spontaneous as suggested by anecdotal evidence" (p. 10-11).

A report by Hawkins (1985) further supports this notion. Two upper elementary teachers introduced Logo to 25 eight- and nine-year-old children and 25 eleven- and twelve-year-olds and observed their development with the programming language over a two-year period. They aimed to gain further understanding of the cognitive and social effects of using Logo. The expectations of the teachers changed during the study. Initially they had hoped that programming would promote problem solving abilities and logical thinking skills. After the first year, they viewed Logo more as "...a context for learning about programming and computers" (Hawkins, 1985, p. 16) than as a context promoting problem solving skills. Transfer of logical thinking abilities were not apparent. In addition, the students' depth of understanding of the language was very low (Hawkins, 1985).

Burns (1983) reports that even though eight- and nine-year-old students at the Bank Street School were able to program in Logo, they did not have a good understanding of the logic of the language because of its abstract nature. Rather, they had a functional understanding only. They could use the commands appropriately but had not internalized the meanings of the commands. Therefore they reached a plateau in their programming ability where they required assistance to write more complex programs.
Despite the opposition to young children's programming and the research which suggests that "powerful ideas" are not being learned as easily as expected, there are many articles in the literature citing examples of young children (of about four years and up—for example, the Lamplighter Project) learning Logo and being successful with it (D'Angelo, 1981). "The Lamplighter experiment does demonstrate... that the rudiments of Logo are easily learnable (and teachable) by very young pupils" (O'Shea and Self, 1983, p. 187). Hines (1983) concluded in her study with five-year-olds that young children can perform simple programming on the computer. Jaworski and Brummel (1984) conducted a study with 28 first graders. The students worked in pairs for 10 weeks with 15 to 30 minutes of computer time each week. She concluded "...the overall hypothesis that elementary age school children of diverse abilities can learn to feel comfortable with and exert some control over the computer was definitely supported by the project" (p. 55). Clements (date unknown) comments about the potential of Logo programming with primary grade children. He states "that young students with high proficiency in programming can perform as well or better than their older counterparts (p. 177). He contends that programming is not beyond the capabilities of young children.

Much of the dispute concerning whether young children can program seems to focus on the definition of programming. Munro-Mavrias (1983) defines programming "as a series of
instructions to a computer in order to have it carry out a task" (p. 1). Papert (1980) defines it as "nothing more or less than communicating to (the computer) in a language that it and the human user can both understand" (p. 6). If these definitions are applied to young children, then this age group is probably capable of programming. If, however, Obrist's (1985) definition is used then the preoperational child probably would not be capable of programming. Obrist contends that "In order to qualify as a program, the sequence of instructions must be written with intent to produce a particular result" (p. 72). In addition, in order to program at this level a child must also be able to use certain problem solving strategies. "The difficulty for a preoperational child in generating problem solving strategies is that she cannot reverse the direction of her thought" (Almy, cited in Burg, 1984). Perhaps, instead of using one definition or the other to define programming, both can be used to define different levels of programming with Obrist's definition representing a higher level of programming than Munro's or Papert's definition. Therefore, many young children using Logo may be in a more rudimentary level of programming rather than the higher level of programming.

Why Logo?

Educators who have chosen to teach young children programming have primarily used Logo as the introductory language. What characteristics of Logo make it appropriate for
early childhood education? When selecting a programming language suitable for young children, Mullan (1984) suggests that one considers the availability of the language, the structure of the language, the usability of the language and the conceptual ease. When these criteria are applied, Logo then appears to be the most appropriate language for young children.

Gillingham (1983) lists two reasons why young children should use Logo. They are that:

1. "Logo offers the raw material to build tools for children's thinking."
2. "Logo offers a flexible and easy to use computer language for exploring one's thinking." (p. 16)

There are numerous other justifications in the literature regarding the suitability of Logo for young children. Birch (1984) states that children can begin using Logo as soon as they recognize the letters of the alphabet since single keystroke programs only use one letter to represent a command (for example, "F" means forward). Papert (1980) refers to the ease by which young children use the language and also to the potential it has for being used to solve complex problems by stating Logo has "no threshold and no ceiling." Another reason cited is that Logo is not too abstract but more concrete and understandable than other programming languages (Clements, Computers in Early and Primary Education, 1985). "Logo is a computer language designed to be accessible to children. It is developmentally appropriate for young children and is interactive" (Clements & Gullo, 1984, p. 1054). Coburn et al. (1985) contends that Logo is less abstract...
than other programming languages. The results of giving commands (drawing of lines and reorientation of the turtle) can be immediately seen on the screen (Clements, *Computers in Early...*, 1985; Coburn et al., 1985).

Marshall (1986) expands on this notion and believes that the Logo philosophy is compatible with the seven goals of the kindergarten curriculum in British Columbia. The goal of emotional development and well-being is promoted in a Logo environment because the successful learning of Logo develops the child's self-esteem and independence. Social development is developed by the sharing of ideas and knowledge that occurs in a Logo environment. Social responsibility is developed because children approach the same problem in different ways so they learn to respect the views of others. The goal of physical development and well-being is promoted through off-computer activities such as practising turtle movements on the floor. Logo encourages aesthetic and artistic development because the students can imagine a picture and create it on the screen through an exploration of the language. Intellectual development occurs because of the nature of the language which requires an interaction of mental and physical action. Finally, language development is fostered because the children learn the commands of the Logo language and must understand their meanings to be successful with programming. They also become familiar with computer related terminology. Marshall believes Logo "is a natural learning process" for the kindergarten child (Marshall, 1986, p. 62).
There are many references in the literature which refer to the term "body syntocity" or "body geometry" as an important factor that Logo is more concrete and easier to learn by young children. Rosser (1983) defines the term as, "bringing new information into harmony or resonance with children's experience and knowledge of their own bodies" (p. 2). Children can relate to the movements of the turtle because they are bodily movements which they have already experienced such as forward, backward, right, and left (Hines, 1983; Martin and Berner, 1982; Pantiel and Petersen, 1984; Papert, 1979; Weir, cited in Jaworski and Brummel, 1984). They transfer their previously acquired knowledge of how their body moves to how the turtle moves.

"Sylvia Weir (1981) reiterated this belief as she felt that as you command the turtle to move around the screen your intuitive knowledge about how your body moves around in space is mobilized" (Jaworski and Brummel, 1984, p. 21).

Papert intended Logo to be self-directed learning where the child is free to investigate, explore and experiment with the language at his own pace. This type of exploration is often referred to as "play" which is a natural means of learning for young children. "The child learns through his or her inner drive to explore, experiment, and discover...As children play, information gained through experience is explored, tested, and represented in a variety of ways" (Ministry of Education, 1984, p. 59). Therefore, children can "play" with the Logo language and become absorbed in the activity since they initiate and control the learning (Maxwell, 1984; Markuson et al., 1983). The concept
of play is of great importance with this age level since, as Chandler suggests, "For the young child, far more learning occurs during play than in any deliberate and structured manner" (Chandler, 1984, p. 9). Play facilitates cognitive development (Ministry of Education, 1984). If Logo creates an environment in which children can play, then the experience should be of benefit to them cognitively. The computer, and more specifically Logo, can be used as "another medium for exploration, expression, and play available to young children and their teachers" (Genishi, 1987, p. 22). Burg (1984) further supports the use of computers in the early childhood classroom and comments, "It's doubtful that microcomputers will ever replace blocks or dolls or trucks or crayons. But the microcomputer can provide children with developmentally appropriate experiences" (p. 31). The computer becomes just another part of a rich environment providing a variety of activities for the young child (Silvern and McCary, date unknown).

Some important characteristics of play listed by Garvey (cited in Chandler, 1984) and Labinowicz (1980) inherent in Logo are spontaneity, voluntary and enjoyable participation, active participation, promotion of enthusiasm, questioning, puzzlement and no extrinsic goals. Besides benefitting from a play experience, Logo programmers also receive immediate feedback regarding the success of their program by watching the execution of the program on the screen (Clements, Computers in Early...
What is Logo?

"Logo is perhaps the best known computer language used in elementary schools today. It was designed specifically for children over a 12-year period by Seymour Papert, his colleagues at Massachusetts Institute of Technology, and other researchers around the world" (Pantiel and Petersen, 1984, p. 79). Logo is a philosophy of education in which the learner is in charge of learning (Jaworski and Brummel, 1984). "Logo can be described as a language created expressly for the purposes of teaching and learning...Thus, unlike many other computing languages created for other than educational purposes. Logo is specifically tailored to meet educational needs" (Upitis, 1982, p. 28).

Papert had a specific type of learning environment in mind when he designed the Logo language. Logo is meant to create "microworlds" or learning environments which "allow children to experiment with ideas by manipulating elements of a computer model in order to discover its structure" (Dewitt, 1984, p. xiv). For example turtle geometry is a microworld which permits children to discover geometric concepts and principles. It is an environment which aims to foster thinking skills. Papert wished to encourage a "top-down problem-solving strategy rather than a "bottom-up" approach (Gollini, 1985). "The top-down programmer starts with a clear model of what he wants to achieve. This is
broken down into the various sub-procedures that are needed and then the sub-procedures solved" (Papert et al., 1979, p. 93). Furthermore, the Logo environment is meant to be a supportive setting in which there is no right or wrong answer. Children experiment with the language to determine a solution to a problem and in so doing "develop (a) scientific approach to learning and experience real problem-solving" (Maxwell, 1984, p. 106). Papert also intended Logo programmers to explore the language independently so they could make personal discoveries about it. Maxwell (1984) believes that these discoveries are possible because Logo "is flexible and so well structured in itself that the children do build on their knowledge as they progress" (p. 106). The student becomes the teacher of the computer and gains a sense of power and mastery (Housey, cited in Jaworski and Brummel, 1984). They attend to the task because they are in control (Jaworski and Brummel, 1984). In addition, children are encouraged to share their ideas and help their peers solve problems in this setting (Willis et al., 1983). In summary, Enns (1986) describes a Logo environment as "safe, discovery-oriented, democratic and epistemological" (p. 30).

The Logo language is based on Piagetian developmental theory. "Piaget's work holds that children are builders of their own intellectual structures and that intellectual development does not always need explicit teaching" (Vaidya and McKeeyby, 1984, p. 30). Children can learn without being taught. Piaget also believes that children learn if they are presented models and examples. Papert used these ideas to create a geometric
learning environment (Eyster, 1981) which uses a discovery or natural learning approach. According to Flake et al. (1985) there are two important aspects of Piagetian theory that are evident in Logo. They are: "1) spatial orientation; and 2) children learn best from their own actions" (p. 24) which both relate to 'body syntocity' or Piaget's term of 'body transfer'. In addition, social elements of Piaget's theory are also evident in Logo. Students are encouraged to share and work with others and also to be creative (Jaworski and Brummel, 1984).

The two learning theories of Piaget and Papert do conflict, however, on certain points. "The Logo approach is underpinned by Piaget... But it is a 'new' Piaget setting aside his framework of natural development in favour of a more 'interventionist' approach" (Davy, 1984, p. 549). Piaget contends that thinking processes develop "according to fixed biological laws, in conjunction with - but never determined by - interactions with the environment" (Riedesel and Clements, 1985, p. 123). Papert (1980) contends, however, that young children will progress through the stages of development at a faster rate if they have Logo experiences. Formal ideas will become concretized through Logo and so understanding of the concepts will occur at an earlier age (O'Shea and Self, 1983; Willis et al., 1983). Children could reach a formal operational level before 11 to 12 years of age. "Papert places emphasis on the intellectual structures that could develop as opposed to those that actually at present do develop in the child" (Papert, 1980, p. 161) which means he believes the stages of development are not rigidly fixed
but can be speeded up. A second point of contention is that Piaget characterizes the five- to seven-year-old as having a topological view of the world rather than viewing the world as having fixed and rigid shapes (Reiber, 1985). Turtle geometry, however, focuses on drawing fixed and rigid shapes.

Underlying the philosophy and learning theory of Logo is a teaching methodology. Since Logo is meant to promote discovery learning, the teacher becomes free to act as a resource person or a facilitator rather than a source for knowledge (Jaworski and Brummel, 1984). The teacher's duties are to guide, to question, to challenge, to clarify, to inspire, to motivate and to facilitate the learning (Reimer, 1985; Streibel, 1983). The teachers "help students gain control over the learning process" (Hortin and Teague, 1983, p. 23). Birch (1984) suggests ways to teach Logo. She advises that, "The questions you ask should be leading questions which guide each child toward a possible solution...Give hints to solutions rather than solutions themselves" (p. 11). Papert intended Logo to foster social interaction between students. Therefore, Collins and White (1984) suggest that teachers should not "hover" over the students but give them enough freedom and encourage interaction between students rather than between student and teacher. In a Logo environment, Riordon (1982) contends that students should decide when they need help, seek that help and therefore be responsible for their own learning. Teachers must be skillful to know when to intervene to assist with problems and when to refrain from interfering (Wellington, 1985).
The literature reveals a difference of opinion about the amount of structure that should be present in a Logo environment. The range extends from a very open setting where the students explore Logo on their own, to a setting in which students select from a variety of tasks, to a setting in which all students must do the same tasks following a rigid series of steps (Kieren, 1984). Mullan (1984) believes that without some structure, the students will not likely discover the more difficult concepts. However, Papert, following true Piagetian learning, does not believe the teacher should provide any structure. He contends that children can learn "powerful ideas" through Piagetian learning which is learning without being taught. "Powerful ideas are ideas that connect deep and important concepts from mathematics, computer science, problem-solving etc., with what is important to children by giving them the power to achieve goals that they see as important" (Leron, 1985, p. 26-27).

Leron does not believe that these two goals--Piagetian learning and powerful ideas--can be attained simultaneously. Students who are left alone to discover with Logo, Leron argues, end up using a "hacking" style of programming. Also, Riedesel and Clements (1985) note that the Progressive Education Movement showed that when children are left in control of their own learning, they limit their learning range. To overcome these difficulties, Leron (1985) recommends a "quasi-Piagetian learning" environment in which the students receive some direction and are given time to reflect on their results. However, students are also given the opportunity to work on their
own projects because "exploration sparked by the student's own imagination and creative energies is central to the Logo philosophy" (Torgerson, 1983, p. 6). Students receive more direction initially and then gradually become more independent (Vaidya and McKeepy, 1983). Clements (Computers in Early... , 1985) also supports the quasi-Piagetian setting and suggests that, "there needs to be a balance between teacher directed lessons, group problem-solving discussions and student-planned and executed projects" (p. 163).

Many others support this quasi-Piagetian style of teaching Logo. "Setting goals and sub-goals does not distract from the philosophy of Logo, it rather enables the child and the teacher to direct their attention to the main objectives, and develop their own way of achieving them" (Mullan, 1984, p. 110). The teacher should teach the commands and provide time for group discussions so that the students may benefit from what their peers have learned. Hawkins (1985) reported an experiment in which two upper elementary teachers taught Logo in their classrooms for a two-year period. The teachers went through a process of interpreting and reinterpreting the value of teaching Logo. One teacher believed that Logo "was a self-expressive medium" so began teaching Logo with little structure. He taught the commands but did not give the students any additional ideas. The second teacher provided new commands when the students were ready. He shared ideas with the students and they shared their ideas amongst themselves. Due to the student's performances, by the end of the study the teachers had changed their somewhat
unstructured views of teaching Logo. "Rather than viewing Logo as discovery-based learning, they saw it as a complex symbol system requiring structured support" (p. 35). The teachers felt that the students did not have a good understanding of the language at the end of the study and that they would benefit from the skills being taught in a coherent sequence.

Kieren (1984), in support of a more structured view, designed a sequence or instructional structure for teaching Logo. He contends that there are definite levels to learning Logo which include:

1. direct mode
2. naive programming mode
3. planned programming mode
4. formal programming mode

The first two levels are in the immediate mode and direct feedback occurs from a particular command but this is not formal programming. The degree of sophistication at these levels depends on the words or commands introduced. The second stage, where preplanning is still not involved, is the beginning of simple programming. The last two levels "represent more insightful programming. These levels demand the ability to make at least simple concrete generalizations and to pre-think the structure of the task" (p. 11). Due to the hierarchical structure of Logo, Kieren recommends moving at a slow pace. He believes that it is, "Better to do sophisticated, successful things at a lower level, than simple but rather rote things at a higher level of language use" (p. 12).
Evaluation is an important part of any teaching methodology. Logo, however, is not meant to be evaluated as other subjects are in the curriculum because it is self-paced and based on discovery learning (Riordon, 1982). Teachers, however, should be aware of the progress their students have made. Birch (1984) recommends the teacher to keep a Logo log book which records types of drawings, approaches to solving problems, repetition of procedures, learning style, and anything unusual about how the student solves Logo problems. Jaworski and Brummel (1984) also suggest keeping anecdotal records and files of the student's work. This type of evaluation enables the teacher to follow the student's development in Logo but does not contradict the Logo philosophy.

Introducing Logo to Young Children

The literature describes several different ways to introduce young children to Logo. This section will examine these methods. A popular suggestion is having the children learn a single keystroke program. There are many software packages available which allow young children to press single keys to draw shapes, designs and pictures and to explore geometric concepts (for example, shape and angle). Children seem to have been successful and the programs effective (Dobson & Richardson, cited in Kieren, 1984; Cathcart & Cathcart, cited in Kieren, 1984; Berdonneau, cited in Kieren, 1984). Clements (1983-84) refers to these programs as "scaffolding" and they enable the children to grasp
concepts that would have otherwise been beyond their abilities. He believes single keystroke programs are necessary. "Without specially-designed assistance using Logo may be either a frustrating experience or little more than graphic fun for many young children... Even with a 'friendly' and accessible language, like Logo, research indicates that full procedural programming is a complex task for children as old as eight or nine years (Pea, Hawkins & Sheingold, 1983). However, if given support in their efforts even preschool children have been reported to create graphics programs" (Clements, 1983-84, p. 24). Leggett (1984) suggests that the threshold for learning Logo may be lowered by providing these "platform programs". Flake et al. (1985) also support the use of single keystroke programs. They reason that young children do not have a good grasp of what large numbers represent but can still benefit from programming using the single keystroke programs. There are many of these programs available on the market and they vary in difficulty.

Two single keystroke programs recommended by Birch (1984) include INSTANT and DELTA DRAWING (this notation symbolizes the registered trademark of DELTA DRAWING). She believes that, "Both of these programs are excellent introductions to Logo at any age and are especially good for young children" (p. 10). DELTA DRAWING introduces the children to procedural programming but in a simpler way. The students do not become frustrated with the complexity of the syntax of the language (Birch, 1984).

Ball (1985) lists some valid reasons for using these types of programs initially with young children. They include:
1. children's limited keyboarding skills do not interfere with cognitive development
2. children lack confidence with two digit numbers
3. total number of commands is small; most of the keyboard is inoperative
4. the number of symbols and associated concepts is small in number and simple in scope (p.374).

The literature presents other reasons as well for using these programs. The children can concentrate on what they want the turtle to create rather than focusing on the verbal aspect of the command (Vaidya and McKeefy, 1984). Children must remember single letters for commands but do not need to concern themselves with correct spellings, spaces and numbers. Another justification for using these programs is that most of the programs produce immediate feedback and visual results as soon as a key is pressed which is unlike true Logo (Tan, 1985). The procedure is executed immediately without the child having to instruct the computer to do so.

Clements (1983-84) cites two reasons why young children have difficulty with programming in true Logo. They are psychomotor skills and thinking skills. Psychomotor skills refer to skills such as typing skills, memory skills, and mathematical skills whereas the thinking skills include such things as planning, ordering, and remembering a sequence of instructions. To overcome these obstacles, Clements designed three levels of support programs for primary students. Level 1 was similar to the many single keystroke programs available on the market but
with a few additions. Children used it to define procedures and build other procedures. Level 2 incorporated "the usual Logo commands but in addition utilized several powerful programs provided by the teacher" (p. 25-26) which helped children when they encountered difficulties. Level 3 used the Logo editor as well as special procedures which were provided. First grade children involved in a three-month study which incorporated these levels demonstrated favourable results.

How long should the children use these single keystroke programs? Birch (1984) suggests that this depends on the individual. "The time a child spends on the introductory material before programming in Logo depends upon the degree of readiness interest and prior experience. Children learn at different rates and should be encouraged to work at a level which they feel comfortable" (p. 11). When students show signs of frustration with a program because it is limiting what they can do, then they should be exposed to a more complex program (Birch, 1984).

Numerous other on and off-computer Logo transition activities are also mentioned throughout the literature. These include:

1. Play Turtle (Campbell, 1985; Jaworski and Brummel, 1984; Martin et al., 1982; Munro-Mavrias, 1983; Reimer, 1985; Torgerson, 1983)
2. Turtle in the Pond (Campbell, 1985)
3. Turtle Talk (Horn, 1986)
4. Masking Tape (Horn, 1986)
5. Milk Carton Forest (Horn, 1986)
6. Turtle Town (Horn, 1986)
7. Simon Says (Horn, 1986)
8. Mr. Robot (Hawaii State, 1984)
9. Gameboard (Hawaii State, 1984)
10. Geometric shapes (Martin et al., 1982)
11. Craft projects (Martin et al., 1982)
12. Cooking experiences

Martin et al. (1982) suggest that these activities are very valuable since they help children expand their experience with Logo and thus increase their learning through Logo. Many of them enable the student to better relate his movements to the movements of the turtle promoting "body syntocity". Others foster procedural thinking skills by requiring the child to follow a sequence of procedures. Careful attention should be paid to the introductory activities and programs presented to young children so that their first experiences with Logo can be positive and successful.

Summary of the Review of Literature

The review of the literature examined many of the issues relating to young children using computers. Studies and opinions both supported and opposed the use of computers with young children. The literature did not present overwhelming evidence to support either side of this issue. Some researchers did not even regard it as an issue. They believe that since computers
are being used with young children, this process will not be reversed. Instead, it is important to determine how the computer should be used rather than whether it should be used.

The next topic of discussion in the review of the literature focused on introducing programming to young children. Once again there are those who favour young children learning to program and those who strongly oppose it. Arguments supporting both sides were presented. Supporters have claimed and some studies have concluded that learning to program can be of benefit to the programmer both intellectually and socially. However, the results of the studies investigating the cognitive benefits were often conflicting. It is still too early to conclusively state that learning to program promotes cognitive growth. Specific areas of intellectual development may improve more than others but studies have not yet clearly determined these areas. The research did suggest, however, that using the computer and learning to program in Logo promoted social interaction between students.

Numerous reasons were cited why programming is too abstract for young children but there is considerable evidence that young children are indeed learning to program at a basic level. If programming can be considered to have different levels then it may not be too abstract for young children.

If young children are going to learn a programming language then it is important that a suitable and developmentally appropriate language is selected. The literature unequivocally supports using Logo with young children. A Logo environment
provides a supportive setting in which children investigate and explore the language to make discoveries on their own. However, some researchers suggest that the Logo learning environment should have some structure to help students make these discoveries and to help prevent stagnation at a particular programming level.

In the final section of the chapter ways of introducing Logo to young children with a focus on single keystroke programs was examined. These programs provide support for young children. They eliminate obstacles which make Logo more difficult for the young child to master. Numerous off-computer activities were also listed. Many of these help students relate the movement of the turtle to their bodily movements. Others promote development of skills necessary for programming.

Based on the literature the focus of the study was refined. The study aimed to further investigate issues about young children learning to program. The language chosen was Logo and it was introduced through a single keystroke program and off-computer activities. Two methods of teaching were selected, a structured approach and a guided discovery approach, to determine how they affected young children learning to program.
Chapter 3

METHODOLOGY

The purpose of the study was to examine the appropriateness of introducing computer programming to kindergarten children. In this study a computer software program called DELTA DRAWING was used. In the study several issues relating to the programming abilities of these children were addressed. The capabilities of the children to program were examined; how the learning environment and teaching technique affected the programming abilities of the children and the acquisition of specific skills were examined; and, the benefits of learning to program were examined. First, this chapter presents the questions relating to these issues. Then there is a discussion of the sample and the design which includes a discussion of the intervention of DELTA DRAWING and the two groups.

In the second part of the chapter the instrumentation is discussed. The statistical component of this study was a mixed-factor design. Pretests and posttests on five different measures as well as a programming posttest were administered to two distinct groups. Details of these tests are presented. The study also contained a qualitative component. Data were collected through observations, a student interview and a parent questionnaire. Information is provided for each of these instruments.
The third section of the chapter outlines the procedure of the study and the last section discusses data analysis.

Questions of the Study

The questions of the study arose from an examination of the literature related to this topic and concerns of the researcher about the subject. Certain issues kept recurring and some of these have been chosen as the focus for this study:

Programming Capabilities of Kindergarten Children

Developmental Issues

1. Can kindergarten children learn to program?
2. Is programming too abstract and complex for kindergarten children?

Suitability of Single Keystroke Program

3. Are single keystroke programs a suitable means for introducing programming to kindergarten children?
4. What are kindergarten students' capabilities in learning to program with single keystroke programs?
Type of Programming Experience

5. Is learning to program a frustrating or rewarding experience for kindergarten children?

6. What difficulties do students encounter when learning to program with single keystroke programs?

Teaching Technique / Learning Environment

7. Is a structured teaching style appropriate to teach kindergarten children to use a single keystroke program?

8. Is a guided discovery approach appropriate to teach kindergarten children to use a single keystroke program?

9. Does the teaching technique affect the ability to program?

10. Does working with a partner affect learning to program?

11. Does working on a computer promote or inhibit social interaction?

Benefits of Programming

12. Does programming with DELTA DRAWING affect children's numeral recognition?

13. Does programming with DELTA DRAWING affect children's upper case letter recognition?

14. Does programming with DELTA DRAWING affect children's understanding of directional / positional terms?
15. Does programming with DELTA DRAWING affect children's recognition of geometric shapes?
16. Does programming with DELTA DRAWING affect procedural thinking skills?
17. Does the teaching technique affect the acquisition of specific skills?
18. Is there an interaction of teaching technique and the acquisition of specific skills?

Sample

Forty kindergarten students participated in the study. The sample consisted of 21 boys and 19 girls. The children, primarily middle-class, attended an elementary school in Richmond, British Columbia. The students attended school for 2.5 hours per day. Sessions were held in the morning and in the afternoon. The 22 morning class students were assigned to the Structured Treatment Group and it was comprised of 12 boys and 10 girls. The 18 afternoon class students were assigned to the Guided Discovery Treatment Group and it was comprised of 9 boys and 9 girls. A convenience sampling was used because random assignment was not possible. In September, class assignments were made on the basis of order of registration. Where possible, the school accommodated parental requests for their child to begin with either mornings or afternoons.
Design

This study had two parts: a quantitative part and a qualitative part. The quantitative part employed a mixed-factor design. Performance was measured by five pretests and posttests and a programming posttest. Data for the qualitative component were collected through observation reports, student interviews and a parent questionnaire. The researcher considered attitudes of the students, interest level of the students, attitudes of the parents, degree of social interaction, and performance on lesson assignments (for the Structured Treatment Group only because the Guided Discovery Treatment Group was not given compulsory assignments).

Intervention

All students in the study used the single keystroke programming language called DELTA DRAWING on the computers. The computer sessions occurred for four weeks between late February and mid-March. Seven to ten computers were in the classroom for these sessions. For both groups there were three sessions a week for three weeks and two sessions the last week. Each session was between 45 minutes and one hour in duration. Both groups received the same number of sessions (11) for the same length of time. The sessions, for both groups, involved group time, on-computer time and off-computer time (students worked on
programming or computer related tasks). During some lessons, the students worked individually on the computers but at other times they worked with partners. The school did not have a computer room so the computers were wheeled on trolleys to the kindergarten classroom at the appropriate time by the grade-seven students. Consequently, the structure of the sessions often varied from what was initially planned because it was affected by the availability or prompt arrival of the computer hardware. Lessons frequently were modified because of problems in one of these areas.

Treatments

This study involved two treatment groups. The 40 kindergarten students were assigned to either the Structured Treatment Group (the morning class) or the Guided Discovery Treatment Group (the afternoon class). The treatments differed by the teaching technique which was employed to present the program DELTA DRAWING. In the review of the literature, opinions varied about the recommended teaching technique for teaching Logo-type programming. Papert, a main proponent of Logo, is opposed to any type of structure when students use Logo. He believes that they should have complete freedom to explore and discover the "powerful ideas" of the language on their own (Papert, 1980). Others believe that some structure is necessary; otherwise, the students will not likely discover the difficult concepts on their own (Leron, 1985; Mullan, 1984). Thus, this
study focused on two different teaching techniques and aimed to compare how the two techniques affected the performance level of the students in using the programming language.

The Structured Treatment Group

The Structured Treatment Group (hereafter referred to as the Structured Group) followed a series of eleven structured lessons about DELTA DRAWING which were presented by the teacher over a one-month period. The teacher used a didactic method of teaching and the students followed the lessons step-by-step. Students had limited choices during the lessons. The teacher assigned both the on- and off-computer tasks. The lessons were carefully planned by the teacher before the study commenced. A scope and sequence for the introduction of the commands were determined and detailed lesson plans were drawn up. Each lesson plan (Appendix H) included the objectives of the lesson, a list of the materials needed, the procedure of the lesson, information about supplementary off-computer activities and details of the assigned project(s). During the course of the sessions a few lessons were modified due to difficulties with previous lessons. The teacher did not insist on moving on to the next lesson without some degree of understanding by the students on the previous lesson. Each lesson for the Structured Group involved:

1. Group instruction. This was given to the whole class by the teacher. During this time new commands were introduced, commands were reviewed, questions were
answered, previous lessons were discussed, supplementary activities were introduced and assigned tasks were presented.

2. Instruction in DELTA DRAWING. This was presented through a series of lessons which progressed through different levels.

3. Supplementary off-computer activities. The activities related to the lessons and were provided for each lesson. These activities included such things as playing turtle, playing games, using gameboards, and discussing geometric shapes. The teacher assigned the students a particular activity for each lesson.

4. Assigned tasks. On-computer projects were assigned to the students for each lesson. Depending upon the outcome of the previous day's project results, the assigned task was sometimes repeated for a second day to improve the mastery level of the task. Some lessons involved both on-computer and off-computer assignments.

A brief summary of each of the eleven lessons for the Structured Group follows.

Lesson 1. Before the lesson started, the teacher used masking tape to form a large (2m x 2m) square, to represent the computer screen on the floor. Then the students were gathered in a large group and the teacher introduced the program DELTA DRAWING to the students. She showed them the "turtle" (cursor) on a computer screen and discussed how they would be learning to control the turtle to move it around the screen. The teacher
explained that the turtle only moved if it was given certain commands. The commands Draw (D), Right (R), Left (L), Move (M) and Erase (E) were then introduced and the teacher used wall charts to assist her explanation. The wall charts had the commands printed on the left side and what the command did (pictorial representation and word) on the right side. The teacher used the computer to demonstrate the effect of each command on the turtle.

Next the students were asked to sit around the masking tape computer screen on the floor. One child was selected to be a "turtle" and asked to stand in the center of the screen. The teacher gave this student computer commands and helped him move in the correct manner around the masking tape screen. Then the students took turns directing the child who acted as the turtle. Students were asked to find a partner. One child in each pair was given the off-computer assignment called Turtle Tracks 1-oc ("oc" referred to off-computer) on paper. There were six parts to the project and it listed the computer commands. For example, the first part listed the following commands: D, D, R, D, L. One student was directed to read the commands while the second student was to act like the turtle following the commands correctly and walking it out on the floor. Once all commands had been read by the first student, the children switched roles.

The students were then gathered together. The teacher explained the computer assignment called Turtle Tracks 1-c ("c" referred to computer). Students were asked to draw lines on the screen in different positions. Eight different lines were drawn
on the assignment sheet. Students were also asked to put the turtle in eight different positions as indicated on the assignment sheet. The teacher assigned the students to partners and then to the computers. Assignments were distributed. Students worked on the computers until the end of the session.

Lesson 2. The teacher read the story The Turtle and the Rabbit to start the lesson. She emphasized how a real turtle moves slowly and steadily and so does the turtle on the screen. Next the five commands introduced in Lesson 1 were reviewed with the teacher referring to the wall charts. Special emphasis was placed on the difference between the commands D and M. Different coloured stickers were put on each child's right and left hands. The teacher directed the students to move about the floor. The students pretended to be turtles and responded to the commands. The right and left commands were focused on during this part of the lesson.

Next the students were gathered together for the teacher to explain the computer project. A sticker was put on the computer screen with masking tape on the back. The children were asked to move the turtle so that it pointed towards the sticker. Students were assigned to partners and computers. Stickers were put on the screens and the students were asked to complete the assigned task. When they finished the task, they put up their hand to indicate to the teacher that they were finished. The students were given sheets of paper with a picture of a turtle on it. Students were asked to put the sticker from the screen on this
sheet of paper and a new sticker was put on their screen. This procedure of receiving a sticker for successfully completed work was referred to as the sticker page system.

Students were gathered on the floor away from the computers. The Turtle Board Game was explained to the class. Small boards (70cm x 30cm) as well as small triangular cardboard turtles (5cm equilateral triangle) were distributed. The children were asked to put the turtle in the center of the board and to locate the nose of the turtle. Students worked with partners. One student moved the cardboard turtle while the second student gave the commands. Students switched roles upon the teacher's direction. The materials were collected and the teacher explained the second computer task.

Stickers were once again put on the screens but the students were asked to move the turtle so that it hid under the sticker. Students were assigned to partners and to computers. The sticker page system was used when the child completed the task. The teacher recorded the total number of stickers that the students received during the computer sessions.

Lesson 3. Prior to the session the teacher made four mazes on the floor using masking tape. Students were assembled and the five basic commands were reviewed once again. The teacher used the wall charts for reference. The students were asked to sit around one of the floor mazes and the teacher explained what to do. One child was selected to be the turtle. The remaining students were asked to give the turtle directions to move him around the maze. Students chose partners and the class was
divided into four groups. Each group was assigned to a different floor maze. Each child in a pair took his or her turn as a turtle and giving the commands. The groups were rotated so that students moved through more than one maze.

Students were gathered and the computer assignment—the maze transparencies were explained. Six different mazes were drawn on six pieces of acetate with a marker. The transparency was attached to the computer screen with masking tape. Students were instructed to move the turtle around the maze. For the example the teacher typed in the commands that the students suggested. Students chose their partners and computers were assigned. Mazes were attached to each computer screen. The sticker page system was used when the students completed the task. They were encouraged to complete more than one maze.

**Lesson 4.** Before the lesson began the teacher made four curved mazes on the floor with masking tape. The commands introduced to the students during the first week of lessons were reviewed by playing a guessing game. The teacher would give a clue about one of the commands and the students had to determine the command. Seven new commands [Colour (C), Hide (H), Redraw (SHIFT G), and scrolling commands (↑, ↓, → and ←)] were introduced. The teacher used the wall charts and the computer to explain these. Next the discussion focused on other types of lines that could be drawn on the computer besides straight lines. Curves and shapes were suggested.
The students were gathered around one of the curved floor mazes and a child was chosen to be a turtle. The remainder of the class was asked to direct the turtle around the maze. A second maze was done with a different child emulating the turtle.

Next the class discussed drawing letters on the screen which would incorporate drawing both straight lines and curved lines. Students were directed to draw their initials on the computer. Partners were assigned and the students worked on this task on the computers. Continuing to work with partners, the children were assigned to a floor maze. Each child in the pair was directed to act as a turtle and to give the commands before moving to a new maze. Children were then reassigned to the computers in pairs and were instructed to continue drawing letters.

Lesson 5. The seven commands introduced in the previous lesson were reviewed. The teacher led a discussion about how to draw curves on the computer. All children were asked to find a space on the floor and the teacher gave the students directions to walk out a curve. This procedure was repeated. The students were gathered and the teacher introduced the letter mazes. The twenty-six upper case letters were individually drawn on pieces of acetate with a marker. A letter maze was attached to the computer screen with masking tape. The students were asked to move the turtle around the letter on the screen. The children were directed to press S when they were finished a letter to save the procedure they had typed into the computer and to put the
turtle back in the center of the screen. Students chose their partners and they were assigned to computers. Since the computers were delayed in arriving, there was no time during the session for an off-computer activity.

Lesson 6. All the commands that had been introduced to date were reviewed by playing a guessing game. The teacher asked questions which began with the phrase, "What do I press if...?" The students had to determine the correct command from the clue. Next the teacher reviewed the letter transparencies which were introduced in the previous lesson. She attached a letter to the screen and had the students suggest how to move the turtle to draw the letter. The new computer assignment called Turtle Tracks 2-c was then introduced. Students were asked to type in the commands as they were printed on the sheet and then to draw the picture which they saw on the screen on their paper.

The "Logo Board Game" was explained to the class. Four students could play the game at once. It reinforced the basic commands of DELTA DRAWING which included D, M, R, L and E. These commands were written on cards. Each player had a turtle marker. A die was rolled and a card was turned over. The player would move his or her turtle the correct number of times in the manner stated on the command card. For example rolling a six and turning over a D card would mean the player moved ahead six spaces. The first person to land on a "Logo" square was the winner.
Half the students were assigned to the computers. They were instructed to complete one letter transparency and one Turtle Tracks 2-c sheet (there were six to choose from). The remaining students were divided into groups of four and played the Logo Board Game. The two groups switched after fifteen minutes. The last ten minutes of the session were spent on the computer. Students worked with partners and chose to do either a letter transparency or a Turtle Tracks 2-c sheet.

Lesson 7. Prior to the lesson the teacher made each of the five basic shapes—circle, square, rectangle, triangle and diamond on the floor with masking tape. The teacher used attribute blocks and pattern blocks to have the students identify the five different shapes and to determine distinguishing features of each shape. Students were then gathered around one of the shapes on the floor. One child was picked to be a turtle and the other students gave the directions to move him around the shape. This procedure was repeated with a second shape. Next the shape transparencies were introduced to the class. The five basic shapes were drawn individually on pieces of acetate with a marker. The shapes were attached to the computer with masking tape. The children were asked to move the turtle around the shape. The sticker page system was reviewed. The off-computer assignment called Turtle Tracks 2-oc was explained. The students had to complete each row by copying the shape at the beginning of each row several times to fill up the space.
The class was divided into two groups. One group was assigned to the computers (children worked independently) and directed to work on the shape transparencies. The sticker page system was used when the students completed a task. Children in the second group were asked to find a partner and move around each of the shapes drawn on the floor. Each pair was asked to take turns being a turtle and giving the directions. Then this group was asked to complete Turtle Tracks 2-oc. The two groups were rotated after fifteen minutes.

Lesson 8. Four new commands [U Turn (U), fill shape (CTRL F), new screen (CTRL N) and erase picture (CTRL E)] were introduced during this lesson and two commands (Save and C) were reviewed. The wall charts were used in the explanation. The five basic shapes were reviewed by playing a game. The teacher provided clues about a shape and the students had to guess the correct shape. Next the shape direction wall charts were introduced. The teacher had printed on large cards the commands needed to draw the five shapes in two different sizes (large and small). There were ten cards in total. These were hung on the wall. The computer assignment called Turtle Tracks 3-c was similar to the wall charts. Each shape page had directions for a shape in two different sizes. Students were directed to type in the commands in the proper order. As each command was typed, students were encouraged to cross off that command on their paper with a pencil. Once the shapes were drawn, the children were asked to fill the shape with colour.
The class was divided into two groups and one group was assigned to the computers. The students were instructed to try to complete at least two shape pages. The sticker page system was employed. The off-computer group was given instructions on Turtle Tracks 3-oc. The teacher gave the students oral directions to colour the shapes on their paper different colours. The two groups switched after fifteen minutes.

**Lesson 9.** The five basic shapes were reviewed with the class by playing a game. The teacher held up one of the five shapes and the students had to hop the correct number of times for each side. Next the teacher reviewed the shape direction cards on the wall. She elicited from the students the patterns that they could see in the directions. The computer project, Turtle Tracks 3-c, was reviewed. The teacher did an example on the computer.

Half the students were assigned to computers and directed to complete two shape direction sheets (Turtle Tracks 3-c). The sticker page system was used when each task was finished. Geoboards and elastic bands were distributed to the off-computer group. Students were instructed to make the four linear shapes in different sizes on the geoboards. The two groups switched after fifteen minutes.

**Lesson 10.** The teacher led a discussion which focused on the similarities and differences of drawing on the computer and drawing with paper and pencil. The teacher then presented four picture files, one at a time, that existed on the DELTA DRAWING disk when it was purchased. Each picture was loaded onto the
computer and discussed with the students. The teacher intended for the students to see the vast potential of the computer program which they were using. Next the teacher used experience chart paper and a marker to draw a face on the chart. She used the students' ideas about how to draw the face and what parts were needed. The teacher provided direction when it was necessary. She then explained the computer assignments which included drawing a face on the computer and then drawing any picture. She recorded the children's intended ideas for pictures.

Half the group was assigned to computers and directed to work on the computer projects. The remaining students were given a choice of using geoboards, pattern blocks, attribute blocks or paper and pencil off the computer. Students who chose to use paper and pencil were asked to draw a picture of something they would like to draw on the computer. The two groups switched after fifteen minutes.

Lesson 11. Prior to the lesson the teacher set up videotape equipment. A student teacher operated this during the lesson. On an experience chart, the teacher drew a picture of something that could be drawn on the computer—a house with trees. She elicited from the students the parts of the picture the order to draw it in, how to draw it and where to draw it. Then the teacher asked the students what they would like to draw on the computer and recorded their thoughts. Half the class was assigned to computers and asked to draw their picture. The
off-computer group could choose to use the geoboards, the pattern blocks, the attribute blocks or paper and pencil. The two groups rotated after twenty minutes.

The Guided Discovery Treatment Group

The Guided Discovery Treatment Group (hereafter referred to as the Guided Discovery Group) had eleven sessions over a one-month period which focused on DELTA DRAWING. The students were given some guidance by the teacher during the sessions, but intervention was limited. The children were encouraged to make discoveries on their own and to share their knowledge with their classmates. Commands were presented on wall charts and the teacher suggested to the students to refer to the charts and to explore and investigate the different commands. At the end of the second lesson, all the commands that the Structured Group would be exposed to were present in this group's environment. The commands were presented on wall charts and posted on the walls. If the teacher deemed the time to be appropriate or the student asked for assistance, individual students were shown, at their computers, how to perform certain tasks. When students asked the teacher for help, the teacher frequently redirected the student to another child who had knowledge in that particular area. If another student was not able to help, the teacher would provide assistance and would ask probing questions to involve the student rather than show the student a formula for solving the problem. During the four weeks of the study the teacher
occasionally created situations to encourage the students to move to a higher level of working with the program. The eleven sessions for the Guided Discovery Group involved:

1. **Group discussions.** These involved both the students and the teacher. The teacher would ask leading questions to guide the direction of the discussion. These sessions presented an explanation of the commands for the programming language, a sharing time of what the children had learned during previous sessions, questions by the teacher to help guide but not dictate the course of the computer time and an introduction and/or an explanation of the supplementary off-computer activities.

2. **Command charts.** The commands for DELTA DRAWING were presented to these students on charts and each command was explained by the teacher. Students used the commands when they chose to do so, if they chose to do so and in any order. Students used free exploration to explore the program.

3. **Supplementary off-computer activities.** A number of these activities (chosen by the teacher) were made available to this group during the non-computer time of the sessions. The students chose an activity from a selected list of activities.

A brief summary for each of the eleven lessons for the Guided Discovery Group follows.
Lesson 1. The students were introduced to the computer program DELTA DRAWING. The teacher showed the students the turtle on a computer screen and discussed how they would be controlling the turtle to move it around the screen. Five basic commands (D, M, R, L and E) were introduced to the students using the wall charts for reference. The teacher demonstrated on the computer.

The children were assigned partners and computers. They were directed to explore the five basic commands on the computer. Off the computer the teacher directed the students to work with a partner. Students could choose from two activities. Students could direct large triangular cardboard turtles (20cm equilateral triangle) attached to straws around the floor by giving it the proper commands. The alternative activity was using the Turtle Board Game (as described in Lesson 2 of the Structured Group Lessons) which involved directing small triangular cardboard turtles around boards. Explicit directions for these activities were not provided by the teacher. She only instructed the children to move the turtles using the computer commands. Students could choose more than one activity during the off-computer time. The teacher gathered the class to discuss what they had discovered during the computer and off-computer activities.

Lesson 2. The teacher discussed all the remaining wall charts and commands. She used the computer to demonstrate how the commands affected the turtle. Children were assigned partners and computers. They were directed to explore and
investigate the commands presented on the wall charts. Off the computer the students could choose from manipulating the large cardboard turtles or the small cardboard turtles. The session ended with a discussion period about what was drawn on the computers and how to use the off-computer materials properly.

Lesson 3. The students briefly discussed some of the different pictures they had drawn on the screen. Then the teacher directed them to think about what else they could draw. Students chose partners and were assigned to computers. They engaged in free exploration on the computers. Off the computer students could use the large and small cardboard turtles or move around the mazes on the floor (used with the Structured Group). Students had two sessions of fifteen minutes on the computer and ten minutes off the computer. The last off-computer time was followed by a short group discussion during which students shared their discoveries of the day.

Lesson 4. Students discussed what they had discovered on the computer during the previous week's sessions. Partners and computers were assigned for the free exploration period on the computers. Off the computer students once again could choose to use the cardboard turtles or the floor mazes. Students had fifteen minutes on the computer, ten minutes off the computer and another fifteen minutes on the computer. There was a brief sharing time at the end of the session. Children shared their discoveries and one child demonstrated what he had learned on the computer.
Lesson 5. The teacher recited a poem about a turtle to get the students' attention and then she taught the students the rhyme. The teacher directed the students' thoughts during the group time. She asked the students: 1) to explain what the wall charts were for and how they could be used; 2) to explain the difference between the commands D and M; 3) if they thought it was possible to draw letters on the screen; and finally, 4) what else could be drawn on the computer screen besides filling it with lines. The students were divided into two groups. One group worked on the computers and were engaged in free exploration. The other group chose from the manipulatives that had been previously introduced to this group as well as the curved floor mazes. The rules of the games were reviewed. The two groups were rotated after fifteen minutes and then the students worked with partners for the last ten minutes of the session.

Lesson 6. Prior to the lesson the teacher drew letters on a few of the screens to see if the children would notice them. There was a brief discussion period during which the rules of the off-computer activities were reviewed and the Logo Board Game was introduced. The class was divided into two groups. One group explored freely on the computers. The maze transparencies (used with the Structured Group) were available for any student who wished to use them. The second group chose an activity from the manipulatives which had already been introduced, the floor mazes
and the Logo Board Game. The two groups switched after fifteen minutes. Then students worked for ten minutes with partners on the computers. The session ended with a sharing time.

Lesson 7. Before the lesson started the teacher drew simple pictures, letters and shapes on some of the computer screens. A group discussion reviewed the results of the previous week. The class was divided into two groups and one group worked on the computers. Shape transparencies (used with the Structured Group) were made available to this group. Students engaged in free exploration on the computers. The teacher asked the students probing questions about their drawings and tried to lead them to a higher level of programming. Off the computer the students selected an activity from the manipulatives, the floor mazes and the Logo Board Game. After fifteen minutes the two groups switched over.

Lesson 8. The results of the last lesson were reviewed during the discussion period. Students explained how to fill shapes with colour and discussed different types of drawings. Half the class was assigned to free exploration on the computers. The teacher reinforced any indications of planned programming. She also suggested to a few students how to combine shapes to make a new picture. The off-computer group selected an activity from the geoboards, the pattern blocks, the attribute blocks, the Logo Board Game and the small and large cardboard turtles. The groups rotated after fifteen minutes. There was a short discussion period at the end of the session.
Lesson 9. The discussion period focused on the variety of pictures that could be drawn on the computer. Students made suggestions about what they would like to draw on the computer. As with previous lessons, there was a computer group and an off-computer group. The computer group engaged in free exploration on the computers with guidance from the teacher. The off-computer group chose an activity from the same materials that were put out in Lesson 8. The two groups switched after fifteen minutes.

Lesson 10. Before the lesson started the teacher loaded a picture file that was on the DELTA DRAWING disk when it was purchased. During group time the teacher presented three other picture files from the program disk. The students inquired about who drew the pictures and about interesting features of the pictures. The group time also included a presentation of a student's drawing of a shark which was completed in Lesson 9. The picture was put on the computer and the child explained it to the remainder of the class. He was given a printout of the picture. The teacher asked the students what they would like to draw on the computer during the session and she recorded their thoughts. Students were then assigned to computer and off-computer groups in the usual manner and the two groups rotated after fifteen minutes. All materials that had been previously introduced were made available to the off-computer group.
Lesson 11. Prior to the lesson the videotape equipment was set up. A student teacher operated it during the lesson. The group discussion provided time for eight students to present to the class their computer pictures from the previous lesson. The class was divided into a computer and off-computer group. The computer group engaged in free exploration and the off-computer group selected from the materials provided in previous lessons. Students worked for twenty minutes and then switched the type of activity.

Instrumentation

Pretests and Posttests

The kindergarten students in both groups were given five pretests and six posttests. They were individually given five pretests and posttests to assess the benefits of learning to program. Four of these tests were BRIGANCE (this notation symbolizes the registered trademark for BRIGANCE) standardized tests of which three were from the BRIGANCE Diagnostic Inventory of Basic Skills and one was from the BRIGANCE Diagnostic Inventory of Early Development. The fifth test, a Procedural Thinking Test, was designed by the researcher. The sixth posttest was a Programming Test which assessed the students abilities to program.
In the review of the literature several studies suggested that children benefit intellectually from learning a programming language. The pretests and posttests aimed to examine this issue for this study. The following tests were administered:

1. Numeral Recognition (BRIGANCE Inventory of Basic Skills).
2. Upper Case Letter Recognition (BRIGANCE Inventory of Basic Skills).
3. Directional/Positional Skills (BRIGANCE Inventory of Basic Skills).
5. Procedural Thinking Test (designed by researcher).

The concepts and skills assessed in these tests were determined by the researcher to relate to the programming language DELTA DRAWING. The pretests and posttests were intended to determine whether there was a change in the level of understanding of the concepts and skills after exposure to DELTA DRAWING.

**Numeral Recognition Test**

For the BRIGANCE Numeral Recognition Test the researcher asked the kindergarten student to identify and name the numerals from one to ten. The numerals were presented in random order on a sheet of paper from the manual for the test. The test was scored
out of ten with one point given to each correct answer. In the student record book, the researcher circled each numeral correctly recognized.

**Upper Case Letter Recognition Test**

For the BRIGANCE Upper Case Letter Recognition Test the researcher asked the student to identify and name the upper case letters of the alphabet. The letters were presented in random order on a sheet of paper from the manual for the test. The test was scored out of 26 with one point given to each correct answer. In the student record book, the researcher circled each letter correctly recognized.

**Directional/Positional Skills Test**

For the BRIGANCE Directional/Positional Skills Test the researcher gave the student directions involving one directional/positional word. The researcher followed the directions in the test manual for what to say to the student for this test. The kindergarten student was asked to perform the correct actions for each word. There were 18 words tested. The test was scored out of 18 with one point given to each correct answer. To score two points for understanding the pair of words "right" and "left" and two points for "forward" and "backward," the student had to answer correctly for both words in the pair. All other words
were scored individually. In the student record book, the researcher circled each word which the student indicated he or she understood.

**Design Concepts Test**

For the BRIGANCE Design Concepts Test the researcher asked the students to identify and name five geometric shapes. The five shapes were circle, square, rectangle, triangle and diamond. The shapes were presented on a piece of paper in the test manual. The test was scored out of five with one point given to each correct answer. The researcher circled correct answers in the student record book.

**Procedural Thinking Test**

The literature also suggested that thinking skills of students improve when they learn a programming language. Consequently, the researcher designed a procedural thinking test to assess programming related thinking skills. The Procedural Thinking Test (Appendix A) was administered as a pretest and a posttest with different forms but the same format for both tests. This test was comprised of three sections which included 1) giving directions, 2) breaking the whole into pieces, and, 3) sequencing. For the "giving directions" section the student was asked to (orally) give the researcher the directions to perform a particular task. The "breaking the whole into pieces"
section involved the student (orally) relating all the pieces that made up the whole—for example, a vase of flowers. The "sequencing" section required the students to put a series of cards (sets ranged from two to six cards) in the correct sequential order. In each of the three sections one test item had two parts, one item three parts and this continued up to six parts. Each item was scored on the basis of the number of parts it contained, so item scores ranged from two to six. There were 15 items in total on the test. Each of the three sections was worth 20 points so the total score of the test was 60. The researcher recorded scores for each test item on a student record sheet (Appendix B) as the test proceeded.

**Programming Test**

The Programming Test (Appendix C) was designed by the researcher to assess the programming abilities of the students at the end of the study. The subjects were asked to produce five predetermined pictures on the computer. The pictures were the same for all students. The items included: 1) surrounding a sticker, 2) drawing an open rectangle, 3) drawing a trapezoid, 4) filling a trapezoid with colour, 5) drawing a lollipop, and 6) drawing a sailboat. The researcher presented pictures of what she wanted produced on the computer screen. There was a time restriction for the completion of each item which ranged from two minutes to five minutes. For each test item, the researcher rated the performance level of the student on a four point
scale. A rating of one was no attempt. Two represented some attempt but showing major problems in the solution. Three represented a good attempt with minor problems in the solution. Four was for total success. The researcher defined the major and minor problems (Appendix D). The researcher recorded the student's score for each test item on a student record sheet (Appendix E). The average of each student's scores was calculated to determine an average total score for each child. The highest possible average score was four.

Other Types of Data Collection

Observation Reports

During the course of the study the researcher kept a journal on both groups. Entries were made for each computer session. The researcher reported on a variety of topics such as the success of the lesson, the interest level of the students, changes which needed to be made for the next lesson, discoveries by students, successes and failures, frustrations of the students and the teacher, cooperation and interaction with other students, and, types of drawings.

After the study, the researcher analyzed the data in the journal to determine trends occurring in the lessons on the various topics listed above.


**Student Interview**

A student interview (Appendix F) comprising of eleven questions was designed by the researcher. The interview attempted to determine the attitudes of the students towards the computer and the programming language, to determine whether the students had an understanding of how the computer program worked, and also, to determine whether the students preferred to work alone or with a partner on the computer. All interviews were tape recorded. The researcher also took notes about the student answers on student interview sheets.

**Parent Questionnaire**

The parents of the subjects were sent a questionnaire (Appendix G) at the end of the study. The questionnaire was intended to assess the level of exposure to computers outside the school. It consisted of 18 questions, but not all of these were required to be answered. Whether a question was to be answered depended upon whether there was a computer in the home and whether the child used the computer. Two questions to assess the attitudes of the parents towards young children using computers and the programming experience the students had in the study were mandatory. Parents were asked to complete the questionnaire within five days and then return it to the researcher.
Procedure

The research study was conducted during a six-week period from the middle of February to late March. Pretesting took place during the first week of the study. The intervention of the program DELTA DRAWING to the two groups occurred during the four middle weeks of the study. Posttesting, administration of the programming test and the student interview took place from the end of the fifth week and throughout the sixth week of the study.

For the pretests, the researcher administered the four BRIGANCE skills tests during one sitting. The tests were given individually to the children during regular classroom hours. Two days at the beginning of the first week were allotted for administration of these tests to all children in both groups. When the student came to the researcher, the researcher explained the general purpose of the tests to the student. For the Numeral Recognition Test and the Upper Case Letter Recognition Test the researcher showed the student pages from the test manual. The Directional / Positional Skills Test involved oral directions by the researcher. The Design Concepts Test also required the test manual. The remainder of the week was used to administer the Procedural Thinking Test. It was administered by the researcher to the students on an individual basis within the classroom and during regular hours. This test involved the teacher explaining orally about each section of the test. Part I and II of the test
required oral responses by the child and Part III required manipulation of sequence cards (prepared by the teacher) by the student.

The lessons proceeded as described in the "Treatments" section above. The Structured Group followed 11 clearly defined lessons over the four week period. The teacher prepared the materials needed for the lessons prior to the sessions. The Guided Discovery Group had 11 lessons over the four week period. These lessons were only structured by the fact that they all had a group time, a computer time and an off-computer time. The computer time and the off-computer time was not structured by the teacher.

For the posttests, the researcher administered the four BRIGANCE skills tests at one time. They were administered at the end of the fifth week. The same procedure was followed for the posttests as for the pretests. The Procedural Thinking Test was once again given separately and was administered at the beginning of the sixth week.

At the beginning of the sixth week the parent questionnaire was also sent out to the students' homes. Parents were asked to return the forms by Friday of the sixth week. Next the Programming Test was given to the students in pairs within the classroom and during regular school hours. It was administered during the middle of the sixth week. The researcher sat between the two children. The computers were situated so that it was difficult for one child to view the other's computer. Each student worked on his or her own computer. The researcher orally
explained to the children what they were expected to do, and they were shown a picture of each item on the test except for filling in the trapezoid with colour. A stopwatch was used to time the students. The student who finished first would wait for the other student to complete the task before proceeding to the next test item.

The student interview was given during regular school hours within the classroom. It was given at the end of the sixth week. Before starting the interview, the researcher discussed the purpose of the tape recorder with the student and then briefly explained the purpose of the interview. Each student was interviewed individually and the interviews were tape recorded.

Data Analysis

The data collected from the four BRIGANCE skills tests, the Procedural Thinking Test and the Programming Test were analyzed using a computer. ANOVA was used to analyze the five pretests and posttests (the four BRIGANCE tests and the Procedural Thinking Test). The computer program BMDP2V ANOVA for a repeated-measures factor and a between-groups factor was used for each test.

The data collected from the Programming Test were analyzed by a computer using a t-test for differences between the two treatment groups. The computer program used was MINITAB.
Part of the data collected from the parent questionnaire was analyzed using a Chi square test of association. The computer program MINITAB was also used to carry out this analysis.

For all tests that were analyzed by computer, computer programs stored at the University of British Columbia Computing Centre were used.

The remaining data from the lesson observations, the student interviews and the parent questionnaires were analyzed by the researcher.
Chapter 4

RESULTS - QUANTITATIVE SECTION

The results section is divided into two parts. Chapter 4 focuses on the quantitative section of the study, the mixed-factor design. The results for the five Pretests and Posttests are presented followed by the results of the programming test. Chapter 5 focuses on the qualitative part of the study. The observations made by the researcher, the results of the student interview and the results of the parent questionnaire are summarized.

Pretest / Posttest Results

Five Pretests and Posttests were administered for this study. The computer program BMDP2V ANOVA for a repeated-measures factor (Pretests and Posttests) and a between-groups factor (the Structured Group and the Guided Discovery Group) was used to analyze all five sets of tests.

Table 4.1 presents a summary of the statistical significance results for the five tests in the study. For each test three outcomes are reported. The first states whether there was statistical significance between the two groups, the Structured and the Guided Discovery, the second states whether there was statistical significance between the two levels of testing, Pretest and Posttest, and the third states whether there was
statistical significance for the interaction of the Group factor and the Test level factor. Following the table, each test is analyzed individually.

Table 4.1
Summary of Statistical Significance for Five Skills Tests

<table>
<thead>
<tr>
<th>Test name</th>
<th>Str. vs G.D.</th>
<th>Pre vs. Post</th>
<th>Interaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numeral Recognition</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
</tr>
<tr>
<td>Upper Case Letter Recognition</td>
<td>S</td>
<td>S</td>
<td>NS</td>
</tr>
<tr>
<td>Directional/Positional Skills</td>
<td>NS</td>
<td>S</td>
<td>NS</td>
</tr>
<tr>
<td>Design Concepts</td>
<td>NS</td>
<td>S</td>
<td>NS</td>
</tr>
<tr>
<td>Procedural Thinking</td>
<td>NS</td>
<td>S</td>
<td>NS</td>
</tr>
</tbody>
</table>

Note. Str. = Structured Group; G.D. = Guided Discovery Group. S = statistical significance at the .05 level or better; NS = no statistical significance at the .05 level.

There was no statistical significance on the Numeral Recognition Test for either the Group factor or the Test level factor. The Upper Case Letter Recognition Test had statistical significance for both the Group factor and the Test level factor. The remaining three tests, the Directional/Positional Skills Test, the Design Concepts Test and the Procedural Thinking Skills Test had statistical significance only for the Test level factor. There was no statistically significant interaction
effects on any of the five tests.

**Numeral Recognition Test Results**

On the BRIGANCE Numeral Recognition Test there was little difference between the means of the Structured and the Guided Discovery Groups. The means of the Pretest and the Posttest scores also showed little difference.

Table 4.2 presents the mean results and the standard deviations for the Structured and the Guided Discovery Groups on the Pretest and the Posttest. Recognition of the numerals between one and ten were tested so the total possible score for this test was ten.

Table 4.2

<table>
<thead>
<tr>
<th>Test level</th>
<th>Structured Mean (SD)</th>
<th>Guided Discovery Mean (SD)</th>
<th>Test level means</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pretest</td>
<td>9.73 (0.88)</td>
<td>9.89 (0.47)</td>
<td>9.80</td>
</tr>
<tr>
<td>Posttest</td>
<td>9.90 (0.29)</td>
<td>9.83 (0.71)</td>
<td>9.88</td>
</tr>
<tr>
<td>Group means</td>
<td>9.81 9.86</td>
<td></td>
<td>9.84</td>
</tr>
</tbody>
</table>

Note. N = 22 for the Structured Group; N = 18 for the Guided Discovery Group.

The ANOVA results for the Numeral Recognition Test are presented in Table 4.3.
Table 4.3

Summary of ANOVA for Numeral Recognition Test

<table>
<thead>
<tr>
<th>Source</th>
<th>Sum of squares</th>
<th>Deg. of freedom</th>
<th>Mean square</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Groups(G)</td>
<td>0.04</td>
<td>1</td>
<td>0.04</td>
<td>0.05</td>
<td>0.82</td>
</tr>
<tr>
<td>Error(G)</td>
<td>26.35</td>
<td>38</td>
<td>0.69</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test levels(T)</td>
<td>0.08</td>
<td>1</td>
<td>0.08</td>
<td>0.73</td>
<td>0.40</td>
</tr>
<tr>
<td>G X T interaction</td>
<td>0.28</td>
<td>1</td>
<td>0.28</td>
<td>2.58</td>
<td>0.12</td>
</tr>
<tr>
<td>Error (G X T)</td>
<td>4.11</td>
<td>38</td>
<td>0.11</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note. N = 22 for the Structured Group; N = 18 for the Guided Discovery Group.

On the Numeral Recognition Test there was no significant difference between the Structured Group and the Guided Discovery Group on test performance. The ANOVA also indicated that there was not a significant difference between Pretest and Posttest scores on this test after the intervention of DELTA DRAWING. The teaching style did not make a difference to learning the numerals to ten. Similarly there was no significant change between Pretest and Posttest scores on the recognition of numerals after DELTA DRAWING had been introduced.
Upper Case Letter Recognition Test Results

The BRIGANCE Upper Case Letter Recognition Test results showed a difference between the means of the Structured and the Guided Discovery Groups as well as a difference between the means of the Pretest scores and the Posttest scores.

The mean results and the standard deviations are presented in Table 4.4. There were 26 letters to be identified on this test so the total possible score for the test was 26.

Table 4.4
Upper Case Letter Recognition Test Means & Standard Deviations

<table>
<thead>
<tr>
<th>Test level</th>
<th>Structured Mean (SD)</th>
<th>Guided Discovery Mean (SD)</th>
<th>Test level means</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pretest</td>
<td>21.63 (5.48)</td>
<td>24.67 (1.97)</td>
<td>23.00</td>
</tr>
<tr>
<td>Posttest</td>
<td>23.32 (4.13)</td>
<td>25.00 (1.88)</td>
<td>24.08</td>
</tr>
<tr>
<td>Group means</td>
<td>22.48</td>
<td>24.83</td>
<td>23.54</td>
</tr>
</tbody>
</table>

Note. N = 22 for the Structured Group; N = 18 for the Guided Discovery Group.

Table 4.5 presents a summary of the ANOVA results for the Upper Case Letter Recognition Test.

There was a significant difference between the test performances of the Structured Group and the Guided Discovery Group on the Upper Case Letter Recognition Test (p = .05). The Guided Discovery Group achieved a higher mean (24.83) than the
Table 4.5
Summary of ANOVA for Upper Case Letter Recognition Test

<table>
<thead>
<tr>
<th>Source</th>
<th>Sum of squares</th>
<th>Deg. of freedom</th>
<th>Mean square</th>
<th>F</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Groups (G)</td>
<td>109.91</td>
<td>1</td>
<td>109.91</td>
<td>4.17</td>
<td>0.05*</td>
</tr>
<tr>
<td>Error (G)</td>
<td>1000.48</td>
<td>38</td>
<td>26.33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test levels (T)</td>
<td>20.10</td>
<td>1</td>
<td>20.10</td>
<td>6.74</td>
<td>0.01**</td>
</tr>
<tr>
<td>G x T interaction</td>
<td>9.00</td>
<td>1</td>
<td>9.00</td>
<td>3.02</td>
<td>0.09</td>
</tr>
<tr>
<td>Error G x T</td>
<td>113.39</td>
<td>38</td>
<td>2.98</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Notes. N = 22 for the Structured Group; N = 18 for the Guided Discovery Group.

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

Structured Group (22.48) on the Posttest but the Guided Discovery Group achieved a higher mean initially on the Pretest. The table also showed there was a significant difference between testing levels for this test (p < .05). The mean of the Pretest (23.00) was significantly smaller than the mean of the Posttest (24.08). The DELTA DRAWING intervention made a significant difference to the ability to recognize upper case letters.

Directional / Positional Skills Test

On the the BRIGANCE Directional / Positional Skills Test there was some difference between the means of the test levels.
but little difference between the group means.

The mean results and the standard deviations for the Structured and Guided Discovery groups on the Pretest and Posttest are presented in Table 4.6. Eighteen directional/positional words were tested for understanding so the total possible score was eighteen.

Table 4.6
Directional / Positional Skills Test Means & Standard Deviations

<table>
<thead>
<tr>
<th>Test level</th>
<th>Structured Mean (SD)</th>
<th>Guided Discovery Mean (SD)</th>
<th>Test level means</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pretest</td>
<td>15.14 (4.07)</td>
<td>15.06 (3.46)</td>
<td>15.10</td>
</tr>
<tr>
<td>Posttest</td>
<td>15.77 (3.90)</td>
<td>15.83 (3.09)</td>
<td>15.80</td>
</tr>
<tr>
<td>Group means</td>
<td>15.45</td>
<td>15.44</td>
<td>15.45</td>
</tr>
</tbody>
</table>

Note. N = 22 for the Structured Group; N = 18 for the Guided Discovery Group.

Table 4.7 presents the outcome of the ANOVA for the Directional / Positional Test.

There was no significant difference between the test scores of the Structured Group and the Guided Discovery Group on the Directional / Positional Test. The teaching technique did not make a significant difference to the ability to understand directional / positional vocabulary. However, the results indicated that there was a significant difference between the Pretest and Posttest scores for this test (p < .05). Once again
Table 4.7

Summary of ANOVA for Directional / Positional Test

<table>
<thead>
<tr>
<th>Source</th>
<th>Sum of squares</th>
<th>Deg. of freedom</th>
<th>Mean square</th>
<th>F</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Groups (G)</td>
<td>0.00</td>
<td>1</td>
<td>0.00</td>
<td>0.00</td>
<td>0.99</td>
</tr>
<tr>
<td>Error (G)</td>
<td>963.80</td>
<td>38</td>
<td>25.36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test levels (T)</td>
<td>9.90</td>
<td>1</td>
<td>9.90</td>
<td>5.37</td>
<td>0.03*</td>
</tr>
<tr>
<td>G X T interaction</td>
<td>0.10</td>
<td>1</td>
<td>0.10</td>
<td>0.05</td>
<td>0.82</td>
</tr>
<tr>
<td>Error (G X T)</td>
<td>70.10</td>
<td>38</td>
<td>1.85</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Notes. N = 22 for the Structured Group; N = 18 for the Guided Discovery Group.

* p < .05.

The intervention of DELTA DRAWING made a significant difference to the performance on this test. After this intervention students performed better on the test of directional / positional vocabulary so the intervention affected performance in this area in a positive direction.

Design Concepts Test Results

The BRIGANCE Design Concepts Test results showed some difference between the means of the Pretest and Posttest scores but little difference between the means of the Structured and Guided Discovery Groups.
The mean results and the standard deviations for the Structured and Guided Discovery groups on the Pretest and the Posttest are summarized in Table 4.8. There were five shapes tested so the total possible score was five.

Table 4.8
Design Concepts Test Means & Standard Deviations

<table>
<thead>
<tr>
<th>Test level</th>
<th>Structured Mean (SD)</th>
<th>Guided Discovery Mean (SD)</th>
<th>Test level means</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pretest</td>
<td>4.36 (0.85)</td>
<td>4.22 (1.06)</td>
<td>4.30</td>
</tr>
<tr>
<td>Posttest</td>
<td>4.68 (0.65)</td>
<td>4.56 (0.98)</td>
<td>4.63</td>
</tr>
<tr>
<td>Group means</td>
<td>4.52</td>
<td>4.39</td>
<td>4.46</td>
</tr>
</tbody>
</table>

Note. N = 22 for the Structured Group; N = 18 for the Guided Discovery Group.

The outcome of the ANOVA for the Directional / Positional Test are presented in Table 4.9.

The Structured Group did not perform significantly different from the Guided Discovery Group on this test. The teaching style did not make a difference in recognizing five geometric shapes. There was a significant difference, however, between the Pretest and Posttest scores on this test (p < .05). This suggests that using DELTA DRAWING made a significant difference in a positive direction to the ability to recognize five geometric shapes because Posttest performance was significantly better than Pretest performance.
Table 4.9
Summary of ANOVA for Design Concepts Test

<table>
<thead>
<tr>
<th>Source</th>
<th>Sum of squares</th>
<th>Deg. of freedom</th>
<th>Mean square</th>
<th>F</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Groups (G)</td>
<td>0.36</td>
<td>1</td>
<td>0.36</td>
<td>0.26</td>
<td>0.61</td>
</tr>
<tr>
<td>Error (G)</td>
<td>51.03</td>
<td>38</td>
<td>1.34</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test levels (T)</td>
<td>2.10</td>
<td>1</td>
<td>2.10</td>
<td>9.52</td>
<td>0.00*</td>
</tr>
<tr>
<td>G X T interaction</td>
<td>0.00</td>
<td>1</td>
<td>0.00</td>
<td>0.01</td>
<td>0.94</td>
</tr>
<tr>
<td>Error (G X T)</td>
<td>8.39</td>
<td>38</td>
<td>0.22</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Notes. N = 22 for the Structured Group; N = 18 for the Guided Discovery Group.

* p < .01.

Procedural Thinking Test Results

On the Procedural Thinking Test there was little difference between the means of the two groups but some difference between the means of the two test levels.

The mean results and the standard deviations for the Structured and Guided Discovery Groups on the Pretest and the Posttest test are presented in Table 4.10. There were fifteen items on the test which varied in value from two points to six points and the total possible score was sixty.
Table 4.10

Procedural Thinking Test Means & Standard Deviations

<table>
<thead>
<tr>
<th>Test level</th>
<th>Structured Mean (SD)</th>
<th>Guided Discovery Mean (SD)</th>
<th>Test level means</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pretest</td>
<td>42.96 (10.40)</td>
<td>43.56 (11.21)</td>
<td>43.23</td>
</tr>
<tr>
<td>Posttest</td>
<td>50.46 (8.57)</td>
<td>49.72 (12.44)</td>
<td>50.13</td>
</tr>
<tr>
<td>Group means</td>
<td>46.71</td>
<td>46.64</td>
<td>46.68</td>
</tr>
</tbody>
</table>

Note. N = 22 for the Structured Group; N = 18 for the Guided Discovery Group.

The ANOVA results are presented in Table 4.11.

Table 4.11

Summary of ANOVA for Procedural Thinking Test

<table>
<thead>
<tr>
<th>Source</th>
<th>Sum of squares</th>
<th>Deg. of freedom</th>
<th>Mean square</th>
<th>F</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Groups(G)</td>
<td>0.09</td>
<td>1</td>
<td>0.09</td>
<td>0.00</td>
<td>0.98</td>
</tr>
<tr>
<td>Error(G)</td>
<td>8011.47</td>
<td>38</td>
<td>210.83</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test levels(T)</td>
<td>924.55</td>
<td>1</td>
<td>924.55</td>
<td>61.96</td>
<td>0.00*</td>
</tr>
<tr>
<td>G X T interaction</td>
<td>8.80</td>
<td>1</td>
<td>8.80</td>
<td>0.59</td>
<td>0.45</td>
</tr>
<tr>
<td>Error (G X T)</td>
<td>567.00</td>
<td>38</td>
<td>14.92</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Notes. N = 22 for the Structured Group; N = 18 for the Guided Discovery Group.

* p <.01.
There was no significant difference between the test performance by the Structured Group and the Guided Discovery Group on this test. The teaching style did not significantly affect the ability to think procedurally. There was a significant difference between Pretest and Posttest performance for this test ($p < .05$). This suggests that DELTA DRAWING made a significant difference to the ability to think procedurally. There was an improvement in test performance after the intervention.

The Pearson correlation coefficient for the Procedural Thinking Test was determined to be .87 where Pretest and Posttest scores for both groups were correlated. The computer program SPSS-X, was used to determine this coefficient.

Programming Test Results

Since the Programming test was only administered after the intervention of DELTA DRAWING a two-sample $t$-test was used to analyze the data using the computer program MINITAB. Table 4.12 relates the means, standard deviations, error means and the outcome of the test.

The results of the $t$-test indicated that there was no significant difference between the performance of the Structured Group on the programming test and the performance of the Guided Discovery Group on the Programming Test. This suggests that the teaching style did not significantly affect the abilities of the students to program.
Table 4.12

Summary of t-test for Programming Test

<table>
<thead>
<tr>
<th>Group</th>
<th>N</th>
<th>M</th>
<th>SD</th>
<th>SE Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Str.</td>
<td>22</td>
<td>3.099</td>
<td>0.475</td>
<td>0.10</td>
</tr>
<tr>
<td>G.D.</td>
<td>18</td>
<td>3.073</td>
<td>0.426</td>
<td>0.10</td>
</tr>
</tbody>
</table>

$t = 0.18$, $p = 0.86$, $df = 37.6$


All students did achieve some level of success on the Programming Test. The student's score was an average out of four. For the Structured Group, the lowest student score was 2.17 and the highest score was 3.83. For the Guided Discovery Group, the lowest score was 2.33 and the highest score was 3.67.

All students made some attempt on five of the six items on the Programming Test. The only test item which some students did not attempt was item four—filling in a trapezoid with colour. Four students in the Structured Group and three students in the Guided Discovery Group were unable to do this task. Table 4.13 presents the results for each item on the test. The number of students in each group for each rating on the scale is given in the item columns.

The two groups found different test items easy and hard. For the Structured Group the items would be arranged as follows: Item 1, Item 2, Item 6, Item 3, Item 5 and Item 4. Item 1 (enclosing a sticker) was the easiest for the Structured Group and Item 4 (filling a trapezoid with colour) was the most
Table 4.13  
Item Analysis for Programming Test

<table>
<thead>
<tr>
<th>Scale</th>
<th>Item 1</th>
<th>Item 2</th>
<th>Item 3</th>
<th>Item 4</th>
<th>Item 5</th>
<th>Item 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>S G.D.</td>
<td>4 3</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
</tr>
<tr>
<td>S G.D.</td>
<td>4 3</td>
<td>8 13</td>
<td>6 5</td>
<td>8 5</td>
<td>12 9</td>
<td></td>
</tr>
<tr>
<td>S G.D.</td>
<td>4 3</td>
<td>10 11</td>
<td>10 9</td>
<td>6 0</td>
<td>4 4</td>
<td></td>
</tr>
</tbody>
</table>

Notes. S = Structured Group; G.D. = Guided Discovery Group.

difficult. For the Guided Discovery Group the items would be arranged in the following order of difficulty: Item 1, Item 3, Item 6, Item 4, Item 2 and Item 5. Item 1 (enclosing a sticker) was the easiest for the Guided Discovery Group and Item 5 (drawing a lollipop) was the most difficult.
Chapter 5

RESULTS - QUALITATIVE SECTION

This section summarizes the observation reports for the Structured Group and for the Guided Discovery Group for each of the eleven lessons and then summarizes the responses to the student interviews and finally to the parent questionnaires.

Observation Reports - Structured Group

Lesson 1. In Lesson 1 the program DELTA DRAWING was introduced to the class and had the students emulate the turtle off the computer. Five basic commands were introduced to the students. Students were assigned the off-computer (oc) assignment, Turtle Tracks 1-oc, where one child read the commands printed on the sheet of paper and a partner walked out the commands. On the computers (c) students were assigned the task Turtle Tracks 1-c. Students made the turtle on the screen draw different lines and had to put the turtle in different positions as illustrated on the assignment sheet.

The lesson did not proceed as the teacher expected. During the computer session, observations of the students indicated that they lacked enthusiasm about the new computer program. Several factors were perceived as contributing to the development of this attitude. Students were paired on the computers. They had been encouraged by the teacher to work cooperatively on the
assigned computer task but only half of the students interacted with their partners while working on the computer. The others worked alone. Consequently, several children waited a long time before they got their turn on the computer. In addition, the assignment on the computer seemed to be too long and overwhelmed the children. They did not persevere and see it through to completion. Six children were off task during the computer session. A final problem with the lesson was that both the off-computer and on-computer periods lasted too long. The students needed more variation. During the off-computer session students were not always moving according to the turtle commands but instead were randomly moving around. Since it was the first day that the commands were introduced, not all students immediately learned the actions corresponding to the commands so some children were not always on task.

Lesson 2. Students were more enthusiastic during Lesson 2 and more children willingly participated in it. The lesson included reviewing the commands which had been introduced in Lesson 1, reading the story The Turtle and the Rabbit, playing turtle, introducing the Turtle Board Game (off-computer game), introducing the sticker page system (students received a sticker to put on a picture of a turtle when the teacher checked a task that was successfully competed) and introducing the two projects for the computer (point to a sticker and hide the turtle under the sticker). There was a variety of activities during the lesson.

The children worked on and off the computers with partners.
On the computer the teacher insisted that the students alternate turns; which they did. This was to avoid one student dominating the computer. The students, however, did not tend to work cooperatively on a project but strictly alternated turns.

All but one student had success with the assigned task. Table 5.1 relates the results for this task.

Table 5.1

Summary of Results for Sticker Task

<table>
<thead>
<tr>
<th>Number of stickers</th>
<th>Number of students</th>
<th>Percentage of students</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>5%</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>9%</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>23%</td>
</tr>
<tr>
<td>3</td>
<td>11</td>
<td>50%</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>14%</td>
</tr>
</tbody>
</table>

Therefore 87% of the students received a minimum of two stickers. This high level of success was reflected in the attitude of the students during this lesson. They were much more positive. Three students had some difficulty getting started on the project and required teacher assistance and encouragement. Two of these three students often lack initiative when working in other areas of the kindergarten curriculum and all three students are afraid to be wrong. They need considerable teacher support to try something new. In this lesson one of these children
became very frustrated and was on the verge of tears when the teacher intervened. The teacher helped her with the commands so that she could be successful. Once the child gained confidence she attacked the problem and was one of the three students to receive four stickers. The second of the three children required some teacher guidance and then she was also successful with the assignment (received three stickers). Towards the end of the lesson the third child still had not had any success with the project. The teacher intervened and the child received one sticker by the end of the period. Two problems with the computer session were: 1) the difficulty the teacher had in checking the students' work on the computer; and 2) the difficulty the teacher had in distributing the stickers quickly.

During the off-computer session the students worked cooperatively with partners on the Turtle Board Game. Each partner had a turn moving the turtle and giving the commands.

Lesson 3. In Lesson 3 the commands introduced in Lesson 1 were reviewed by having the students move the turtle around a maze on the computer (drawn on a transparency and taped to the screen) and by giving appropriate commands to a partner so that the other person would move through a maze on the floor (drawn with masking tape). For the computer session the interest level of the students was high and they were keen and enthusiastic. One child was even heard to say, "Wow! This is fun!" The children worked with partners once again but took turns as before. Partners were of their own choice. The sticker page system was used so stickers were awarded to students once
they had successfully moved the turtle through the maze on the screen. All students received at least one sticker. Table 5.2 presents the results.

Table 5.2
Summary of Results for Maze

<table>
<thead>
<tr>
<th>Number of stickers</th>
<th>Number of students</th>
<th>Percentage of students</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>36%</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>14%</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>45%</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>5%</td>
</tr>
</tbody>
</table>

The students were quite successful on this task. Sixty-four percent of the students received two or more stickers. On the off-computer mazes the children completed the mazes but there were some behavioural problems because initially the rules for moving between mazes were not clearly established.

Lesson 4. The second week of instruction in DELTA DRAWING began with Lesson 4. New commands (C, H, SHIFT G, ↑, ↓, →, and ←) for the computer were introduced and drawing curves and letters were discussed. The students were asked to make their initials on the screen. Off the computer, students were directed to move around curved mazes, which were made out of masking tape on the floor. Generally the class found this lesson quite difficult. The expectations for the project seemed to be too
high for the age level. During the computer session nine children were not on task and during the off-computer session thirteen students were not on task at some time. Enthusiasm during this lesson waned. The students had most difficulty with the curves. On the computer their frustration level was high when they could not make a curve properly and consequently could not make their initial. One of the children who had difficulty in Lesson 3 was once again very frustrated and wanted to give up. The teacher encouraged her and helped her persevere. Not all students expressed a negative attitude towards the lesson. One child who experienced some success when he completed the first letter in his name was heard to utter, "Yeah!"

Lesson 5. In Lesson 5 the teacher continued instruction on drawing curves and letters on the screen but used a different teaching strategy. Letters were drawn on transparencies and the children chose a letter to tape to their computer screen. The students were directed to move the turtle around the letter. This technique enabled the students to have considerably more success than in Lesson 4. Not all students completed a letter but they stayed on task and made a good effort to move the turtle around the letter. Student programming did not tend to be accurate. Many children's drawings wandered from the path of the letter. Students would try to bring the turtle back on course but they usually did not erase their mistakes. During the lesson the students required a lot of direction but they did stay on task. Two students who had experienced difficulty in Lesson 3 were successful in this lesson and were very excited about their
accomplishments. Once again, however, the children had difficulty working together. In several instances one child dominated the computer. For example, at one computer, one child completed two letters but she had not given her partner a turn on the computer. All but five students remained on task during the off-computer game where the teacher gave the students directions for moving like the turtle. The off-computer activity was cancelled because the computers arrived late.

Lesson 6. For Lesson 6 the class was divided into two groups. After the large group time, one half of the students worked on the computers. Each student had his own computer. The remaining students played a new game called the "Logo Board Game". Two tasks were assigned to the computer group. First the students were to complete a letter transparency and secondly to type in commands from a command sheet (Turtle Tracks 2-c) and draw the results on paper. The groups switched after fifteen minutes and during the last ten minutes the students worked with a partner on the computer doing one of the assigned tasks. The children were very task oriented during the lesson. The results for the letter project are shown in Table 5.3.

On this task 90% of the students experienced success. The two students who did not complete one letter were two of the students who had trouble with previous lessons. Table 5.4 gives the results of the second computer assignment.

Seventy percent of the students on this assignment experienced some degree of success. Two of the students who had difficulties in other lessons sought the teacher's assistance.
Table 5.3

Summary of Results for Letter Project

<table>
<thead>
<tr>
<th>Number of letters</th>
<th>Number of students</th>
<th>Percentage of students</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>10%</td>
</tr>
<tr>
<td>1</td>
<td>12</td>
<td>60%</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>25%</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>5%</td>
</tr>
</tbody>
</table>

Note. Two students were absent for this lesson.

Table 5.4

Summary of Results for Command Sheets

<table>
<thead>
<tr>
<th>Success on command sheet</th>
<th>Number of students</th>
<th>Percentage of students</th>
</tr>
</thead>
<tbody>
<tr>
<td>No attempt</td>
<td>2</td>
<td>10%</td>
</tr>
<tr>
<td>Not correct</td>
<td>4</td>
<td>20%</td>
</tr>
<tr>
<td>Completed 1</td>
<td>12</td>
<td>60%</td>
</tr>
<tr>
<td>Completed 2</td>
<td>1</td>
<td>5%</td>
</tr>
<tr>
<td>Completed 3</td>
<td>1</td>
<td>5%</td>
</tr>
</tbody>
</table>

Note. Two students were absent for this lesson.

One of these children experienced success with both computer projects while the other still was not successful on either project. During the off-computer session children worked in groups of four and were on task.
Lesson 7. The third week of instruction began with Lesson 7. The lesson focused on geometric shapes and how to draw them on the computer. This session was a very quiet working session where there were only three instances of off-task behaviour. Two of these three situations involved students who were sharing a computer. For the computer assignment, students were given shape transparencies which were taped to the screen and asked to make the turtle go around the shape. Students chose the shapes and the teacher encouraged them to complete a minimum of two shapes. Two tasks were assigned for off the computer. First, students were asked to work with a partner to move around masking tape shapes on the floor and then there was a worksheet on which the students drew various shapes (Turtle Tracks 2-oc).

On the computer task, students had most difficulty with tracing circles. Their tracings strayed from the shape and were not accurate. Instead of alternating drawing and turning, the students would draw for too long or turn the turtle too much. The results for the on-computer task are given in Table 5.5.

The student who did not complete one shape worked diligently at tracing a circle. The tracing did not remotely resemble a circle so the teacher asked him to repeat the task but he did not have time to complete the second attempt before the session ended. For the six children who had to work with partners on the computer the results are presented in Table 5.6.

All students in this group had some level of success which was not affected by working with a partner. Students worked
Table 5.5
Summary of Results for Shape Transparencies

<table>
<thead>
<tr>
<th>Number of shapes</th>
<th>Number of students</th>
<th>Percentage of students</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>5%</td>
</tr>
<tr>
<td>1</td>
<td>7</td>
<td>32%</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>36%</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>18%</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>9%</td>
</tr>
</tbody>
</table>

Table 5.6
Summary of Results for Students with Partners on Shape Transparencies

<table>
<thead>
<tr>
<th>Number of shapes</th>
<th>Number of students</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

quickly through the shapes and exhibited a high level of interest and excitement. They enjoyed being successful. One of the students who had considerable difficulty and was frustrated with previous lessons experienced success in this lesson. She completed one shape and seemed disappointed when she had to leave the computer and was not able to complete a second shape. For the off-computer task three students did not do the worksheet.
Another two students did not complete the sheet. All others successfully finished the task. For the turtle emulation game of moving around shapes on the floor, students were focused and on task. Providing two off-computer activities helped to prevent behavioural problems which arose previously.

Lesson 8. Geometric shapes were also the topic for Lesson 8 but new tasks were assigned. Students were given a sheet of paper with instructions for drawing a shape in a small size and a large size (Turtle Tracks 3-c). They had to type in the commands for the shape. They chose the shape they wished to draw. Table 5.7 presents the outcome of this project.

Table 5.7
Summary of Results for Shape Directions (1)

<table>
<thead>
<tr>
<th>Number of shapes</th>
<th>Number of students</th>
<th>Percentage of students</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>14%</td>
</tr>
<tr>
<td>1</td>
<td>13</td>
<td>62%</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>24%</td>
</tr>
</tbody>
</table>

Note. One student was absent for this lesson.

Even though most of the students completed at least one shape, only five drew the selected shape in two different sizes. Since a comparison of sizes and how it affected the instructions was an important part of the lesson, the teacher decided to...
repeat the project in the next lesson. The teacher was needed to assist several students to perform the task. She helped them type the correct keys. Although the students did not experience overwhelming success with this lesson, they persevered and made an effort to do the assignment. The off-computer assignment (Turtle Tracks 3-oc) required the children to colour different shapes according to teacher directions. This was a simple project for the children and all completed it properly except for two students who did not do the task. The teacher had a second off-computer activity prepared but there was not sufficient time for it to be implemented.

Lesson 9. The shape direction sheets introduced in the previous lesson were reinforced in Lesson 9. The computer project from the last lesson was repeated. The sticker page system was employed. Off-computer students used geoboards to make linear shapes in different sizes.

The expectations of the computer assignment were better understood by the students and they applied themselves to the task quicker. The outcome of this assignment is given in Table 5.8.

These results indicated more success on Lesson 9 than on Lesson 8. In the former lesson, 76% of the children completed zero or one shape directions and 24% finished two shape directions. For Lesson 9, however, 33% completed zero or one shape directions while 67% finished two or three shape directions. For the off-computer activity, students were directed to make geometric shapes on geoboards. Most students
Table 5.8

Summary of Results for Shape Directions (2)

<table>
<thead>
<tr>
<th>Number of shapes</th>
<th>Number of students</th>
<th>Percentage of students</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7</td>
<td>33%</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>48%</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>19%</td>
</tr>
</tbody>
</table>

Note. One student was absent for this lesson.

followed these directions and then proceeded to make other pictures such as designs. Throughout the lesson the attitude of the children was quite positive.

Lesson 10. The final week of computer instruction in DELTA DRAWING began with Lesson 10. The computer assignment involved drawing a face on the screen and once this was completed the child could draw any picture. With the aid of a chart, the teacher helped the students determine the parts necessary for drawing a face and a possible sequence for drawing the parts. The teacher drew a face on the chart. Off the computer the students could choose from a variety of materials which included geoboards, pattern blocks, attribute blocks and paper and pencil.

The most difficult part of the computer drawing for the children was making the outline of the face. A few students made square or rectangular shaped faces. Several others needed the teacher's suggestion to follow the directions on the large circle shape direction chart (posted on the wall) to start their face.
Only seven children satisfactorily finished the face task. Of the seven, five began a drawing of their own choice. One of these children completed a picture of a person. A child who was easily frustrated in other lessons was once again ready to give up because she had trouble fitting a second eye inside the face but with teacher encouragement she persevered. The off-computer task was to create a picture using pattern blocks or attribute blocks or alternatively to draw a picture, on paper, of something that they would like to draw on the computer. Six children chose the latter task while the others chose the building task. All students stayed on task.

Lesson 11. The final instructional session, Lesson 11, gave the students some choice. They drew whatever they wished on the computer. The teacher recommended that the students break the whole picture into parts. She did an example by drawing on an experience chart and then on the computer to illustrate this point. Students were required to tell their ideas to the teacher before working on the computer so that she could record their plans.

Three children did not pursue their ideas and randomly played with the keys. Thirteen students attempted to draw what they had planned. Five students used the pictures that they had drawn the previous day on paper to guide their drawings on the computer. One student was absent. The teacher gave assistance to several children on how to get started. One child in particular needed considerable guidance from the teacher. She had not attempted to start the project but also had not sought
the teacher's help. She knew what she wanted to draw but was unable to devise the procedure to do it. The variety of pictures on the computer suggested that the majority of the students had learned that with the proper instructions the computer can create any picture. A few of the students had advanced to the stage where they used preplanning as an aid to their computer drawings. The off-computer activities were the same as for Lesson 10 and the children had no problems in this area. There were very few incidents of children not on task in both the computer and off-computer group.

Summary of Observation Reports for the Structured Group

The observation reports for the Structured Group reveal some important points about this group. Although they had a frustrating experience during the first lesson, the students did not develop a negative attitude towards the computers and this initial frustration did not negatively affect the remainder of the sessions. Their interest level and enthusiasm seemed to depend on the particular structure and expectations of each lesson. When students experienced some degree of success during the lesson, they usually had a positive attitude and persevered on the task. When expectations were too high for their level, they became frustrated and somewhat negative. The projects that they had most difficulty with were drawing curves and letters and drawing a face. All of these tasks involved drawing circles or parts of circles. The students discovered that it is much easier
to draw straight lines on the computer than curves. Student drawings during the last session indicated the variety of levels within the group. Some children were at the random hitting stage, while others tried to draw planned pictures and a few were trying to replicate pictures that they had previously drawn.

Students frequently found working with a partner a frustrating experience. One student would often dominate the computer time so that the other child had no turn or a limited turn on the computer. At times the partner situation was beneficial because the students would draw on each other's knowledge to complete the task.

Observation Reports - Guided Discovery Group

Lesson 1. The Guided Discovery Group was introduced to five basic commands during Lesson 1. The teacher explained the meaning of each command and used the wall charts to assist in the explanation. Then partners were assigned and the children worked in pairs on the computers. The teacher asked the students to explore and investigate the five commands to see what happened to the turtle. When the children worked on the computers, they began to randomly hit the keys rather than choose a particular key to explore. Consequently, they discovered numerous commands other than the five introduced on the chart.

The general feeling of the session was one of overwhelming enthusiasm and excitement. The teacher found it difficult to keep up with the demands of the students and did not have enough
time to observe all the children. They were continuously asking her to come and see their screen. Many comments indicated the enjoyment the students were getting from the lesson. Some of these included:

Wow! See my lines.
Look at mine. Come here.

One child who was very keen on the computer demonstrated a desire to plan what he was doing. He asked the teacher for assistance and asked questions such as, "I want to go here and then there. How do I make it go up?"

Not all students, however, were enthusiastic about the new computer program. There were two children working together who were of low ability and found it difficult to take the initiative and discover things about the program. They quickly became frustrated and started to blame the software for their difficulties. One of the children said, "How do you make it do anything? Ours isn't working." The teacher and their peers helped them to get started.

Most of the children remained on task for the computer session. Two children excelled during the session in terms of their enthusiasm and their investigation of the program. Five children lost interest during the session. One of these five was quite enthusiastic at the beginning of the lesson but then his interest waned. Another child was very focused for approximately twenty minutes but then became upset because his partner was not sharing the computer. The remaining students
applied themselves to the task. The students frequently interacted with the teacher and their peers throughout the session sharing newly acquired knowledge. Many of the children also worked alone for long periods of time.

The off-computer session was not as successful. The teacher presented two types of manipulatives to emulate the movement of the turtle. The students chose which manipulative they wanted to use. During this session there were several instances of off-task behaviours. Materials were not being used properly. Many students did not use the correct commands to direct the turtles. A few children, however, did work well in this session.

The discussion period allowed a few children to present what they had discovered during the on- and off-computer sessions. Some children simply related what they had drawn while others were able to explain the outcome of a particular command. After the first lesson, the teacher felt confident that the students would make progress in this group without too much teacher intervention.

Lesson 2. The remaining programming commands were introduced to the class on charts during Lesson 2. After the introduction of these commands, this group had all the commands that the Structured Group would be learning over the eleven lessons. All command charts were hung on the wall, close to the computers, for easy reference. Hardware problems slowed the momentum of excitement and enthusiasm that had begun in Lesson 1. The computers arrived quite late. Only seven of ten computers arrived and two computers malfunctioned during the
session. Partners were assigned and two groups had to work as a triad.

During the computer time, the children in the larger groups, in particular, had difficulty taking turns. Otherwise the lesson went satisfactorily. Students continued to make discoveries and ask the teacher for assistance or ask their peers for assistance. There was a variety of different types of interactions. Students worked alone, students observed their peers without conversation, students helped each other, and students sought the teacher's assistance. The off-computer activity, which involved the same manipulatives as the previous lesson, was more successful than Lesson 1 because the expectations of this period were better understood by the students.

Lesson 3. Lesson 3 started and ended with brief discussion periods. In the opening discussion the teacher guided the students to talk about what they had done previously and then to discuss what else they could try on the computer. The teacher helped the students summarize their discoveries. The explanations provided by the students suggested that they were gaining an understanding of various different commands. Children worked in pairs on the computers. Some worked cooperatively while others had trouble sharing. Working on a colour monitor became important to the children during this lesson because they began to use the colour command. Students were explaining to others how to draw in colour on the computer. The teacher encouraged children to ask their peers for help as well as to come to her for help. The sharing of information became very
evident during this lesson. It was apparent between partners but also between groups working on different computers.

There was some off-task behaviour during the computer time. One child consistently did not use the computer. This child needed considerable direction and reassurance by the teacher in other activities at school. She was unable to take the initiative to try different commands on the computer so she did not participate.

There was some off-task behaviour during the off-computer session, but basically the students stayed on task. During this session children directed cardboard turtles about but they frequently did not state the commands aloud. The teacher reminded the class that the turtle on the computer was unable to move without commands. Students began to make a transfer between the turtle on the screen and the cardboard turtles which they moved about on the floor. (For example, one child during the off-computer time indicated that he had assimilated what the command K, four turtles appear on the screen, did on the computer when he said, "I can't do K. Then I'd have to be four.") Although the children continued to make discoveries like these, the teacher was concerned about the amount of random key hitting that continued on the computer.

Lesson 4. The second week of the study began with Lesson 4. During the discussion period the teacher guided the group to recapitulate what was learned during the first week. Students were then assigned computers. Off the computer students chose to use the cardboard turtles or the floor mazes. A couple of
children drew pictures on the computer during this session. One was a diamond and the other was a dragon. The child who drew the diamond was able to explain how she did it. She was one of two children at this stage who had made the transition from the random hitting stage to planning what they were doing. The dragon, on the other hand, was a random occurrence. Once the lines were drawn, the children involved in the drawing decided that it looked like a dragon. The teacher was impressed during this session about the degree of cooperation on the computers. Students did not tend to work independently but instead were interacting with their partners and with others. The colour monitors had become of such importance to the students that one child refused to work on a black and white monitor. He finally did so after persuasion from the teacher. Both on and off the computer there were some incidents when children did not apply themselves to their tasks but the majority remained quite focused.

Lesson 5. The teacher started Lesson 5 with a discussion during which she asked leading questions to help motivate the children to move to a higher level of working with the programming language. The group was divided into two. One group worked with partners on off-computer tasks which included two types of turtle manipulatives (large and small cardboard turtles) and floor mazes. Students selected the activity they wanted to use. The computer group engaged in free exploration and worked individually on the computers. After fifteen minutes the two groups switched. On the computers, the teacher reinforced and
praised students who had not cluttered their screens with random lines or who had made close approximations of letters. From the computer drawings, it was evident that three students had now moved from random hitting to more planned programming.

Off the computer, the children had considerable trouble staying on task and required a lot of teacher direction. Two students, however, were doing a good job of shouting out commands and moving around the mazes. Another two children hugged each other when they successfully moved around a curved maze. The teacher praised these two groups. For the last ten minutes of the session the children chose a partner and they worked together on the computers. Four children made letters and two children made a man. The other drawings were random. In these paired situations some children dominated the computer and did not give the other child much or any opportunity to participate.

Lesson 6. For Lesson 6 the teacher decided to set up a situation to try to help more children make the transition from random hitting to more planned programming. She drew letters on a few computer screens before the session began and waited to see if any children noticed them. With this minimal amount of guidance many children started to draw letters, numbers, shapes and pictures. Some of the pictures included a block letter "X" which was filled with colour, venetian blinds (a random drawing and then the child named it), a kite, and coloured rectangles. Far fewer students cluttered their screens with random lines.

Children interacted well with others and were drawing upon the knowledge of their peers to perform certain tasks. Many
students requested the teacher's attention simply for recognition of their accomplishments. Others required assistance but the teacher would usually redirect the child to another child who knew how to perform the desired skill. During the discussion period children were more than willing to describe their drawings. One child shared how to make a circle. She had discovered the technique on her own. In the last computer session of the day, when the children were paired, twelve students had a purpose to their drawings. On the computer all children were on task except the child who lacked self-confidence. During the off-computer session the Logo Board Game was introduced and all the other manipulatives (previously introduced) were available as well. Most students stayed on task. The teacher had been concerned that the students were not going to progress from the random hitting stage but the results of this lesson disproved this concern. Many students were progressing to a higher level of programming.

Lesson 7. The first lesson in the third week of the study was Lesson 7. Prior to the session, the teacher drew simple pictures as well as numbers and letters on the computer screens. When the students went to the computers they did not appear to notice these drawings because they did not comment about them. During the class, six children decided on what they were going to draw on the computer. Filling shapes with colour was a popular choice of this group. Students who knew how to do this shared their knowledge with others. The girl, who had been reluctant to explore on her own, discovered this technique from
another child and began to apply it. She was thrilled with her achievements. Other students were not yet combining shapes to make more complicated pictures even though the teacher was encouraging them to do so. The teacher asked questions such as, "Can you turn this shape into something else? What can you add to it? What does it look like?" She did not insist that the students draw more complex pictures but she did demonstrate to a few individuals how to combine shapes to make a picture. All students were on task on the computer and there was a minimal number of off-task behaviours in the off-computer group. Two students discovered an off-computer assignment in the classroom that had been assigned to the Structured Group and asked for the instructions which they were readily given. They both completed the task successfully. Other children used the turtle manipulatives, the floor mazes and the Logo Board Game during the off-computer time.

Lesson 8. Before the next scheduled session began, six children came to the computers which were set up in the classroom. They began to use the DELTA DRAWING program. They were keen to work on the computers. During the discussion period of Lesson 8, several children discussed the pictures that they had drawn in the last lesson. Some of these included a "P", a flag and a rectangle. The rectangle was drawn by a child who followed the directions from a shape direction chart which was hung on the wall and had been used with the Structured Group. During the session, the teacher continued to ask questions which might lead the students to higher level programming on the
computer. More of the children began to draw planned pictures. The child who followed the directions for the rectangle in Lesson 7 repeated this task and then followed the directions for a circle. One other child attempted the circle but was not successful. New manipulatives, geoboards, pattern blocks and attribute blocks, were introduced for the off-computer session. The other manipulatives which had been used previously were also available. All students chose to use the geoboards and were quite interested in the activity. After a period of time, one child grew tired of these and selected the pattern blocks.

**Lesson 9.** By Lesson 9 seven students had moved to the planned programming stage. One child in particular drew a much more complicated picture than any of the other children. He drew a picture of a shark with fins and a tail. He worked diligently on his picture and it was quite intricate.

![Computer drawing by a kindergarten student of a shark.](image)

**Figure 5.1**

Computer drawing by a kindergarten student of a shark.
This drawing was another major breakthrough for this group because it indicated the potential complexity of the drawings to the other students. The programmer presented and talked about his picture to the remainder of the group. Other children continued to draw simple pictures while the remainder of the class were still at the random hitting stage. One child had decided to draw a spoon but gave up on the project because it had not worked. She did not consider erasing the parts that she did not like. During this lesson both the on- and off-computer periods were very quiet but busy. Students selected from the same materials that were made available in Lesson 8 for the off-computer period.

**Lesson 10.** The first lesson of the final week was Lesson 10. It began with a presentation of a printout of the picture of the shark to the student who had worked on this project. Then the teacher loaded picture files which were on the program disks onto the computer. These pictures were of sailboats moving along the water, a sun, and an easter egg. The children were extremely interested in these pictures and asked several questions concerning their source. The teacher then asked the students what they would like to draw on the computer and she recorded their thoughts.

During the computer session, nine children worked on discernible pictures but they were not always the same thing as what they had stated to the teacher during group time. The teacher was disappointed that only three children drew pictures of what they had initially planned. The nine pictures included a
sailboat, a motorboat, a guitar, a snowy mountain (the mountains were coloured white) a sun, an egg, a book, an M, a triangle and a treasure chest. The child who worked on the sailboat wanted to replicate the picture he had seen on the screen at the beginning of the lesson and kept asking the teacher questions such as, "Did yours have a sun and clouds?" When he finally finished his picture, he was ecstatic with his work and threw his hands in the air to express his joy. Another child began to work on a project but became quite discouraged when he realized his computer did not have colour.

Interactions between the students were good and so were student-teacher interactions. The children asked the teacher for help and sought her attention to show her their work. No behavioural difficulties arose during the computer or off-computer sessions. Off the computer students chose an activity from the materials previously introduced. Most children used the geoboards to draw pictures. One child drew a picture on the geoboard similar to the one that she drew on the computer (the mountain and the sun). Other pictures included a motorboat, a person and a house.

Lesson 11. The last lesson for the Guided Discovery Group, Lesson 11, was videotaped. Pictures from the previous lesson were presented in a brief discussion period and then students were assigned to either the on- or off-computer group. All materials previously introduced were made available to the off-computer group. On the computers, nine children drew pictures but only four were of objects that had been
predetermined. The pictures included an elephant, boats, a rabbit, a shark and shapes.

From observations of the students' drawings, it was evident that the children had mastered numerous commands during the eleven sessions. Even the girl who had lacked confidence to try out the program on her own initially was drawing shapes and with the help of other children filling them with colour. She was even able to give the teacher a detailed explanation of how to draw a square on the computer and to describe the commands needed to do this. Independent of teacher intervention, the students began relating the pictures on the computer to the pictures on the geoboards. During this last session, the students attended to their activities both on and off the computer.

Summary of Observation Reports for the Guided Discovery Group

The observation reports for the Guided Discovery Group revealed some important points. Through the observations over the eleven sessions, it was evident that the many children had mastered some or all of the following commands: D, R, L, E, T, G, C, S, K, SHIFT G, CTRL F, CTRL E, and CTRL N. The commands were incorporated into their drawings. Also the students' vocabulary during the lessons suggested that they had gained an understanding of various commands. For instance, one child commented, "We have to save it." Initially, the teacher had been concerned that this group would not make considerable progress on their own. This skepticism proved to be wrong very early in the
eleven sessions. The students discovered many of the commands and they were able to explain what happened when they pressed them.

Teacher intervention was kept to a minimum during the sessions. Guidance rather than intervention was the teacher's goal. The discussion periods were valuable in providing an opportunity for the teacher to provide some guidance and suggestions to the students without dictating the course of the lesson. This time also enabled the students to share with their peers their findings and drawings.

Since minimal teacher intervention was the goal of this group, the teacher encouraged the students to work cooperatively and share information with each other. This strategy worked with this group and the students used each other as well as the teacher for assistance.

Throughout the eleven sessions, the student interest level was basically very high. The majority of the group remained enthusiastic and excited. Students enjoyed sharing their achievements with the teacher and their peers. A couple of children had difficulty with this type of learning possibly because of their personality traits. They prefer or need to have guidance and direction rather than working independently to make discoveries on their own.
Student Interviews

This section summarizes the general outcomes of the student interviews.

The student interview had four purposes:

1. To determine the attitudes of the students towards the computer and the programming language.
2. To determine whether the students had some understanding of how the computer worked.
3. To determine whether the students preferred to work alone or with a partner on the computer.
4. To determine what the students enjoyed or found difficult to do on the computer.

Table 5.9 summarizes the children's responses concerning their attitudes towards the computer. One question on the interview focused on this topic and it asked, "Do you enjoy using the computers? Why or why not?" The majority of students had a positive attitude towards the computer.

Table 5.9
Student Attitudes Towards the Computer

<table>
<thead>
<tr>
<th>Group</th>
<th>Positive</th>
<th>Negative</th>
<th>Undecided</th>
</tr>
</thead>
<tbody>
<tr>
<td>Str.</td>
<td>20</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>G.D.</td>
<td>16</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

Note. Str. = Structured Group; G.D. = Guided Discovery Group.
Three questions concerned the students' attitudes towards programming on the computer. Table 5.10 summarizes the responses to two of these questions. Question 4 asked, "Did you like to use the computer to draw? Why or why not?" and Question 9 asked, "Do you like pictures drawn using the computer? Why or why not? What do you like or dislike about the pictures?"

Table 5.10
Student Attitudes Towards Programming

<table>
<thead>
<tr>
<th>Group</th>
<th>Positive</th>
<th>Negative</th>
<th>Undecided</th>
<th>Not understand</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Str.</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>G.D.</td>
<td>15</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>No. 9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Str.</td>
<td>18</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>G.D.</td>
<td>15</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Note. No. 4 = Question 4 of the student interview; No. 9 = Question 9 of the student interview.

Once again the majority of students in both groups had a positive attitude towards the program DELTA DRAWING. The results for the remaining question about this topic are presented separately. Question 5 asked the students, "What would you like to do more often: a) draw with a computer; or b) draw with a crayon and paper? Why?" Table 5.11 summarizes the responses.
Table 5.11

Drawing Preference - With or Without Computer

<table>
<thead>
<tr>
<th>Group</th>
<th>Computer</th>
<th>Percent</th>
<th>Crayon &amp; paper</th>
<th>Percent</th>
<th>Both</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Str.</td>
<td>11</td>
<td>52%</td>
<td>7</td>
<td>33%</td>
<td>2</td>
<td>10%</td>
</tr>
<tr>
<td>G.D.</td>
<td>9</td>
<td>53%</td>
<td>6</td>
<td>29%</td>
<td>2</td>
<td>12%</td>
</tr>
</tbody>
</table>

Both the Structured Group and the Guided Discovery Group had more students who preferred to use the computer to draw than to use crayons and paper to draw. Some of the reasons given by the students included the following:

It's (the computer) easier. When you fill things in, it doesn't go outside the lines.
You have to make the turtle move.
Because you can get it on a printer.
You can make fill-ins. (fill in a shape with colour)
You don't make as many mistakes with the computer.
You don't have to get the crayons and pencils and stuff.
Because the computer is faster.
(The computer is) more fun. You have keys on them.
(The computer is) easier.
It's much funner. You get to press whichever button you think is right. You don't get to press buttons when you draw pictures. You just colour.

The last reason stated was from a girl who had experienced considerable difficulty with some of the lessons but this statement suggests that she enjoyed the computer experience.

Some of the reasons that students preferred to draw with crayons and paper included:
I like to draw pictures and they're nice. It's not very hard. Computers are hard. You have more colours. You wouldn't have to do all the hard work. It's easier to draw a circle and that. It's faster to do. (You can) do more colours than the computer. It's easier than the computer. You just colour it instead of thinking which keys you have to do. You can just colour.

The majority of these reasons focused on the difficulty of using the computer to draw. Many of the children found it a more mentally demanding task than drawing with a crayon and paper.

The student interviews also helped to determine whether the students had some understanding of how a computer works. The relevant question asked, "How do computers draw pictures?" Some of the responses to this question were:

By buttons. You press them. The turtle draws a picture and makes a line. You press the controls and then you get a picture. Push D and it goes up. Not just up. It draws something. You push a button and the turtle moves. By pressing the keys. Then you get a picture. By pushing the keys. It leaves a line behind it. You can fill it in with a colour. You press the keys and they help you. You tell them. Press the buttons. It makes the turtle go. By controlling the turtle. By pressing the buttons. They need a screen and buttons. If you press D you get a line. If you press C you get different colours.

The answers indicated that most students had a basic understanding of how to make the turtle move. Keys had to be pressed and this caused different movements of the turtle. A few
students realized that they had the control over the computer and the computer responded to their directions. Two students gave unrealistic answers to this question which were:

- They have workers. Things that control the turtle for moving lines.
- Push a magic button. It makes a good picture.

Both responses suggest that the students perceived the computer as having some mystical powers which enabled it to draw pictures.

One question asked the students about how they liked to work on the computer. The question asked, "Do you like working alone or with a partner when you use the computer?" Table 5.12 summarizes the responses to this question.

Table 5.12
Preference Towards Working Alone or with a Partner

<table>
<thead>
<tr>
<th>Group</th>
<th>Alone</th>
<th>Percent</th>
<th>Partner</th>
<th>Percent</th>
<th>Both</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Str.</td>
<td>10</td>
<td>50%</td>
<td>9</td>
<td>45%</td>
<td>1</td>
<td>5%</td>
</tr>
<tr>
<td>G.D.</td>
<td>13</td>
<td>72%</td>
<td>4</td>
<td>22%</td>
<td>1</td>
<td>6%</td>
</tr>
<tr>
<td>Total</td>
<td>23</td>
<td>61%</td>
<td>13</td>
<td>34%</td>
<td>2</td>
<td>5%</td>
</tr>
</tbody>
</table>

A much higher percentage of students in the Guided Discovery Group preferred to work alone than in the Structured Group. The Structured Group was almost evenly divided between working alone and working with a partner. When the total responses are
analyzed, ignoring the groups, the majority of students preferred to work alone on the computer.

Reasons stated by the students for working alone included:

I don't have to be bothered.
It's fun. I get to be by myself. I get to learn by myself what to do.
I can work by myself.
Because I always get my turn.
Because I can do it everytime by myself.
You can get it done faster.
You don't have to get off the computer again and again.
You can do whatever you want.
You can do things more often.
You can think.
Because they (partners) do it wrong. They don't know my plan.
Some people share and some people don't.
I can't concentrate.
I can make my own things. I can have a lot of fun by myself.
The other person doesn't mess up your picture.

The main points that the students made in these responses were:

1. The students wanted to be on the computer continuously and not interrupted.
2. The second person was distracting.
3. The time spent on the task was perceived to be faster when working alone.
4. Some students had difficulty sharing.
5. The partner did not always understand the other person's project.

Some of the reasons that students cited for working with a partner included:
(The) partner helps me.
(I like my) friends with me.
I like company.
You can ask for help.
If I do something wrong, he helps me.
It's easier. We each press the keys.

The main focus of the majority of these responses was the helping capacity of the other person. Another point was the companionship that the partner provided.

The last concern of the student interview was what the students found enjoyable or difficult to do on the computer. Student responses varied considerably. Responses to the question which asked, "What is your favourite activity when we use the computer? Why is it your favourite activity?" included:

A rainbow. I like to make rainbows.
A circle. It's like a face.
A boat. I like boats.
Drawing a lollipop. I liked drawing sticks and lollipops on the computer.
Making faces. I can make different things on the face like square eyes.
Doing the boat thing....You had to do a lot of hard work.
Draw a shape and fill it in.
Drawing a square because I wanted to make a square on the computer.
The boat because it was sailing. It moved.
The paintings - fill it (a shape) with colour. I never knew how to do it and then I did it.
Making an Easter basket because I made a picture of it at home and then I did it.
The shapes - the circle and the triangle. They're easy.

The answers given to this question covered a broad spectrum of ideas. Most of them focused on particular objects or shapes that students enjoyed drawing. One student enjoyed a procedure that
he had learned to do (filling a shape with colour) on the computer. Another child was pleased with her ability to draw the same picture on paper as well as on the computer. This student had reached the planned programming stage.

At times, students did have difficulties on the computer. The question which was concerned with this topic asked, "What do you find hard to do on the computer? Why is this hard for you?"
Some of the answers were:

Making a flower. It was hard to do the leaves. (An) Easter egg. I tried to make it round.
Faces because it's hard to do the circle.
Pressing the buttons. Things happened that I didn't want to.
Drawing the circle. The turtle would go in and wouldn't touch the other line.
The boat. Too hard to make the flag on top of it.
The shapes - rectangle and square.
I can't make a circle. I tried to make a circle and I made a square.
Making a fill-in flag. I didn't know how to make a fill-in and that was hard.
Filling the pictures in. Sometimes you couldn't close them up and get the turtle in.
Drawing a man. I can't draw the eyes.
The shapes - the ones that have pointed sides and the circle because I didn't do them before so I didn't really know how to.
When I did shapes - the diamond. It had points and I couldn't figure out which way to go.
Writing your name.
The S. You have to go like a real S on a piece of paper. (It's) easier on paper.

Common problems for the students included drawing the various shapes, and particularly circles, carrying out specific procedures (for example fill-ins) and drawing letters. However, the difficulties were quite specific and no one stated that
everything was difficult to do on the computer.

In general the student interviews suggested that the students had a positive attitude towards the computer and the program DELTA DRAWING, they had gained a basic understanding of how the computer worked, they preferred to work alone on the computer and at times the computer tasks were both enjoyable and difficult.

Parent Questionnaire

Parent questionnaires were sent out to all student's homes at the beginning of the last week of the testing period. Parents were asked to complete it and return it to school within five days. Of the 40 questionnaires distributed, 28 were returned.

One question inquired whether there was a computer in the child's home. Table 5.13 indicates the number of students with or without computers in the home and the number of students using the computer at home.

Table 5.13
Computers in the Home and Computer Use by Children

<table>
<thead>
<tr>
<th>Group</th>
<th>Quest.'s returned</th>
<th>Computers</th>
<th>No computers</th>
<th>Home use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Str.</td>
<td>17</td>
<td>8</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>G.D.</td>
<td>11</td>
<td>4</td>
<td>7</td>
<td>4</td>
</tr>
</tbody>
</table>
All parents who returned the questionnaire answered two general questions regarding their views about the issue of young children using computers and whether the computer experience at school had been positive or negative for their child. Data related to the answers for Question 17 are summarized in Table 5.14. Question 17 asked, "Do you support or oppose young children under eight years old using computers? Explain your answer." Two types of answers were given which included responses supporting the use of computers and mixed responses where the respondent listed both supporting and opposing reasons for using computers.

Table 5.14 indicates that there were more reasons stated by parents which supported the use of computers with young children than reasons stated which opposed their use.

The second general question asked on the questionnaire was Question 18. The question asked, "Do you think the experiences your child had with computers through this study was positive or negative? Explain your answer." The data summarizing the responses to this question is given in Table 5.15. Once again there were two types of responses given. One type of response described indicators that the experience was positive. The other type of response was a mixed response where both positive and negative indicators were mentioned.

As Table 5.15 indicates, parents cited indicators that suggested the experience was positive more frequently than they cited indicators that suggested the experience was negative.

The remaining questions on the questionnaire were not
Table 5.14

Responses Supporting or Opposing Young Children Using Computers

<table>
<thead>
<tr>
<th>Reason</th>
<th>No. of quest.'s</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Become familiar with a technology which will be an integral part of the child's life as an adult.</td>
<td>14</td>
</tr>
<tr>
<td>2) Provides a stimulating and challenging educational environment.</td>
<td>8</td>
</tr>
<tr>
<td>3) Beneficial to start as early as possible.</td>
<td>6</td>
</tr>
<tr>
<td>4) Reason not stated but responded in favour of using computers.</td>
<td>3</td>
</tr>
<tr>
<td>5) Easier to learn skills at young age.</td>
<td>2</td>
</tr>
<tr>
<td>6) Fear of computers does not develop.</td>
<td>2</td>
</tr>
<tr>
<td>7) Promotes letter recognition.</td>
<td>1</td>
</tr>
<tr>
<td>8) Promotes eye-hand coordination.</td>
<td>1</td>
</tr>
<tr>
<td>9) Provides an accomplishment with an immediate reinforcement.</td>
<td>1</td>
</tr>
</tbody>
</table>

Mixed Responses - Supporting Reasons:

1) Important skill for the future.                                      2
2) Fear of computers does not develop.                                  1
3) Enhances education.                                                  1
4) Beneficial to start as early as possible.                            1
5) Reason not stated.                                                   1

Mixed Responses - Opposing Reasons:

1) Computer skills are not a substitute for basic skills.               3
2) Children should not play computer games at expense of other subjects. 1
3) Valuable time is wasted because the children lack typing skills.     1
4) Fear of radiation.                                                   1
5) Computer is not important at this age.                               1
Table 5.15
Indicators of Positive or Negative Computer Experience

<table>
<thead>
<tr>
<th>Indicator</th>
<th>No. of quest.'s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive Indicators:</td>
<td></td>
</tr>
<tr>
<td>1) Child was excited about computers.</td>
<td>11</td>
</tr>
<tr>
<td>2) Child more aware of computers.</td>
<td>4</td>
</tr>
<tr>
<td>3) Child more comfortable with home computer.</td>
<td>4</td>
</tr>
<tr>
<td>4) Child wants a computer/ computer games.</td>
<td>3</td>
</tr>
<tr>
<td>5) No reason stated.</td>
<td>3</td>
</tr>
<tr>
<td>6) Good opportunity for students to be exposed to computers.</td>
<td>2</td>
</tr>
<tr>
<td>7) Computer experience of benefit for higher grades.</td>
<td>2</td>
</tr>
<tr>
<td>8) Computer aids educational and creative processes.</td>
<td>2</td>
</tr>
<tr>
<td>9) Computers promote skill development e.g. letters, numbers.</td>
<td>1</td>
</tr>
<tr>
<td>10) Child liked printout.</td>
<td>1</td>
</tr>
<tr>
<td>11) Computer helps develop child's confidence.</td>
<td>1</td>
</tr>
<tr>
<td>Mixed Responses - Positive Indicators:</td>
<td></td>
</tr>
<tr>
<td>1) Child liked some parts of the study.</td>
<td>2</td>
</tr>
<tr>
<td>Mixed Responses - Negative Indicators:</td>
<td></td>
</tr>
<tr>
<td>1) Child found some tasks difficult.</td>
<td>1</td>
</tr>
<tr>
<td>2) Child has not asked to purchase program.</td>
<td>1</td>
</tr>
</tbody>
</table>

answered by all parents. Some questions were answered by parents with computers in their homes, some by parents where the child used the computer in the home and some by parents whose child used a Logo-type graphics program.

Posttest scores of children who had a computer at home and used it were compared with posttest scores of children who did not have a home computer. A Chi square test of association was
used to determine whether using a home computer affected posttest scores. Table 5.16 presents the observed (O) and expected (E) counts and the Chi square results.

Table 5.16
Chi Square for Home Computer Group vs. No Home Computer Group

<table>
<thead>
<tr>
<th>Test</th>
<th>Computer</th>
<th>No computer</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>O=9</td>
<td>O=28</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>E=9</td>
<td>E=28</td>
<td></td>
</tr>
<tr>
<td>Numeral Recognition</td>
<td>O=6</td>
<td>O=23</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>E=7</td>
<td>E=22</td>
<td></td>
</tr>
<tr>
<td>Letter Recognition</td>
<td>O=8</td>
<td>O=22</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>E=7.3</td>
<td>E=22.7</td>
<td></td>
</tr>
<tr>
<td>Directional/</td>
<td>O=7</td>
<td>O=22</td>
<td>29</td>
</tr>
<tr>
<td>Positional Skills</td>
<td>E=7</td>
<td>E=22</td>
<td></td>
</tr>
<tr>
<td>Design Concepts</td>
<td>O=8</td>
<td>O=22</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>E=7.3</td>
<td>E=22.7</td>
<td></td>
</tr>
<tr>
<td>Procedural Thinking</td>
<td>O=7</td>
<td>O=24</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>E=7.5</td>
<td>E=23.5</td>
<td></td>
</tr>
<tr>
<td>Programming</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Chi square = 0.44, df = 5, N = 40

The Chi square test results are not significant which suggests that there was no association between owning a home computer and performance on the Posttests. Students who did not own and use a home computer performed as well on the Posttests as the students who owned and used a home computer.

From both groups only one child used a Logo graphics program outside of the school setting. This child had used the program
for less than one month and he usually worked with an adult on the program. The questionnaire stated that he liked to draw circles when using the program at home. However, looking at the child's results on the section of the programming test which involved drawing a circle (the lollipop test), he received a ranking of two (major problems with drawing). The group mean for this question was 2.3. Thirteen students scored two on the scale and five students scored three on the scale. Working with the program at home did not seem to affect his programming at school. The lesson observation reports, however, indicated that this child was the first to draw a detailed and complicated drawing which was a picture of a shark.
Chapter 6

CONCLUSIONS, IMPLICATIONS AND RECOMMENDATIONS

This study was designed to investigate the appropriateness of introducing computer programming to kindergarten children. The questions of the study focused on three major areas. Questions 1 to 6 dealt with issues regarding the capabilities of kindergarten children to program. Questions 7 to 11 dealt with issues concerning the teaching technique and the learning environment. Questions 13 to 18 dealt with issues about the benefits of learning to program. This section summarizes the conclusions of this study for each of the three areas. Then, implications for the classroom and recommendations for further research in this area are discussed.

Conclusions - Programming Capabilities of Kindergarten Children

Developmental Issues

Munro-Mavrias (1983) defines programming, "as a series of instructions to a computer in order to have it carry out a task" (p. 1). Obrist (1985) states that, "in order to qualify as a program, the sequence of instructions must be written with the intent to produce a particular result" (p. 72). Applying Munro-Mavrias' definition of programming to the outcome of this study, the conclusion is that the kindergarten children were
engaged in programming. The programming test results indicated that all students were able to instruct the computer to perform specific tasks that they wished it to carry out. Students were able to give the computer a series of instructions to produce a desired result. If writing of the commands is considered a higher level of programming than simply pressing the appropriate keys to achieve a specific result, then the students in this study were not programming at the higher level but they were engaged in a basic level of programming.

These findings oppose Burg's (1984) belief that young children are unable to give the computer a set of instructions to do a particular task "even in the most simplest [sic] case" (p. 32). The students in this study did not necessarily know the complete set of instructions before they set out on a task but they were able to put together a series of commands to produce a particular result.

The programming test results for both the Structured Group and the Guided Discovery Group indicated that learning to program was not too abstract or complex for the kindergarten students in this study because some level of success was achieved by all children. All students made some attempt on five of the six items of the programming test. The only test item where students made no attempt was on item four—filling a trapezoid with colour. Three students in the Guided Discovery Group and four students in the Structured Group were unable to do this task. On all the test items, only 18% of the students experienced no success (and this was on item four). In addition, out of a total
possible average score of 4, students in the Guided Discovery Group achieved a class mean of 3.07 with the lowest average score of 2.33 and the highest average score of 3.67. In the Structured Group, the class mean was 3.10 with the lowest average score of 2.17 and the highest average score of 3.83. This level of success on the programming test suggests that kindergarten students are indeed capable of programming. This finding further supports Hines' (1983) results which found that six kindergarten children were able to perform simple computer programming. It also supports the results of the study by Jaworski and Brummel (1984) and the results of the Lamplighter Project (Watt, 1982) which both found that young children were capable of programming or controlling the computer.

The lesson observations and the student interviews further supported the findings of the programming test. Not only were the students capable of directing the computer to carry out a task but they also gained some understanding of how the program worked and how the turtle moved. Most students were able to give a realistic explanation that the keys controlled the turtle and that different keys resulted in different movements of the turtle. Their ability to verbalize what was happening on the screen suggested that they had internalized what they had learned about how the turtle moved. They had achieved a basic understanding of what was happening. If programming had been too abstract and complex for the children they may not have been able to provide a reasonable explanation.
Some of the tasks set out for the Structured Group in a few of the lessons appeared too complex for the students at the kindergarten level. However, when the teacher modified the task, the students always became more successful. There was never complete failure by everyone in the group on any lessons, which suggests that in general the goal of teaching kindergarten students to program was not unrealistic and that the concepts and skills introduced were generally not too complex. The teacher changed the lesson plans to accommodate the needs of the students and to better suit their abilities. The students in the Structured Group did not reach the editing and debugging stage, which the teacher had initially intended. In the Guided Discovery Group it was evident that some students proceeded from the random hitting stage to the planned programming stage to the preplanned programming stage but no one reached the editing and debugging stage in this group either. Perhaps, students did not reach this higher level of programming, the editing and debugging stage of programming, because they were not developmentally prepared. Their logical thinking skills may not have been sufficiently developed to program at this level. This editing and debugging stage may be beyond the abilities of kindergarten children. However, if the study had been extended over several more weeks then some students may have reached this level. This study was unable to make any conclusions about whether kindergarten children can learn to edit and debug programs.
The results of this study suggested that the capabilities of kindergarten students in learning to program was highly individual. Students' performance levels on assigned tasks for the Structured Group, on personal projects for the Guided Discovery Group and on the programming test varied greatly. In both groups some children mastered a vast number of commands while other children had a limited repetoire of commands.

In the Structured Group, individual student performance ranged from no success on some assignments to total success. In the Guided Discovery Group, individual drawings ranged from random hitting of keys to drawing pictures on paper and replicating it on the screen. All students did, however, achieve some degree of success on the programming test. All students had learned some commands to direct the turtle around the screen. Several factors may have affected the level of programming achieved by the students. The interest level of the students and the amount of self-motivation may have contributed to their ability to program. While there were no group differences in programming performance between the Structured Group and the Guided Discovery Group, individual performance may have been influenced by the teaching technique. Some students' style of learning may have been better suited to a different learning environment. The intellectual level of the child may also have affected the level of programming the student could attain. Future studies need to address these issues.
This study determined that kindergarten children are capable of programming and that all children in this studied mastered the use of some commands. The highest level that these students achieved (in both groups) was the preplanned programming stage where children drew a picture on paper and then recreated it on the computer. Since the computer sessions on DELTA DRAWING ended after eleven lessons in one month, it is unknown whether the students would have continued to progress to higher levels of programming if they were given more time.

Suitability of Single Keystroke Programs

The results of this study support the popular view stated in the literature that a single keystroke program is a good vehicle for introducing programming using Logo graphics to young children (Cathcart & Cathcart, cited in Kieren, 1984; Clements 1983–84; Dobson & Richardson, cited in Kieren, 1984; Flake et al., 1985; Leggett, 1984).

The lesson observations for the Structured Group indicated that by the second lesson 95% of the students experienced success with the assignment. Students directed the computer to carry out desired tasks very quickly. The high level of success at an early stage would not likely have been possible if the programming language was too complicated. A possible reason that students were successful almost immediately was due to the simplicity of the program. In the early sessions of the Guided Discovery Group, many children were engaged in random hitting of
keys and lines and designs covered the screens. It was difficult to ascertain from this situation the suitability of the single keystroke program. However, the students did seem to immediately realize that pressing different keys would make something happen on the screen. If the program had been more complicated there would not likely have been as many drawings on the screens. Due to the simplicity of the program, the students could randomly press keys and produce a drawing. However, the program did not seem to be too easy because no students reached the editing and debugging stage of programming and the students did not utilize all of the commands available in the program.

The programming test results also suggested that since all students were capable of programming at some level then the programming language was suitable for the age group. No student experienced complete failure during the lessons and on the programming test. Some students required teacher assistance with determining appropriate keys to press but all students at the end of the month were able to do some programming independently.

The student interviews indicated that an overwhelming majority of students had a positive attitude about the programming experience. If the programming language had been too difficult for this level, the students would not likely have been so positive about their experience.

In both groups most students did not seem hindered by the syntax of the programming language. The students did not have difficulty inputting the commands but instead had difficulty remembering which key (letter) represented which command. Wall
charts helped overcome this problem. Some students had trouble remembering the set of instructions for a particular procedure but all students were able to use some commands. The syntax of the programming language, DELTA DRAWING, seems much simpler than Logo. Logo requires the programmer to state the length of line segments and the size of angles whereas DELTA DRAWING allows the programmer to do this if desired but otherwise uses default values for length and angle size. In addition, DELTA DRAWING only requires the students to press a maximum of two keys (for example CTRL F) for any command. Logo requires several keys to be pressed (for example FD 10). The results of this study suggest that the single keystroke program DELTA DRAWING is a suitable means for introducing programming using Turtle graphics to kindergarten children.

Type of Programming Experience

The study showed that learning to program was primarily a positive experience for the kindergarten children involved in the research. The student interviews indicated that the majority of students had a positive attitude towards both the computer and towards programming in DELTA DRAWING. Almost all students liked to use the computer to draw and liked pictures that were drawn by the computer. A majority in each group (52% of the Structured Group, 53% of the Guided Discovery Group) and an overall class (both groups) majority (54%) said they preferred to use the computer to draw than to use crayons and paper. The reasons for
this choice focused on ease of use, the capabilities of the computer (for example fill-ins and printouts), speed and enjoyment. Some of the students found the computer to be an enjoyable alternative to drawing on paper. It was a new art medium for drawing.

The parent questionnaire further suggested that the computer experience had been positive for the children. The parents cited more positive indicators about their children using the computers than negative or mixed responses. Of the 28 questionnaires returned, 11 parents described their child's enthusiasm and excitement about using the computers at school. If the students had not been enjoying the experience, they would not have expressed this kind of emotion to their parents. Numerous other positive indicators were also related by the parents. Only two parents of the 28, cited negative indicators about the computer experience. One parent related that the child had found some tasks difficult. This is a legitimate reason to develop a negative attitude towards the experience but it is worthy to note that the parent chose to use the word "some" and not "all". This implies that the entire experience was not negative because the child experienced some success. The other negative reason cited was that a child had not asked to purchase the program DELTA DRAWING for their home computer. The parent based the child's enjoyment of the program on wanting to buy the program. For a number of reasons, the child may not have asked to buy the program but it does not necessarily mean that he did not have a positive experience on the computer.
The lesson observations present more varying attitudes towards the computer and programming than the student interviews or the parent questionnaires. This may be due to the fact that the lesson observations were more specific. Student and class activities and attitudes were recorded for each session. The student interview and the parent questionnaire, however, presented a more general view based on the whole study. It is unrealistic to think that every session for every child was positive. The lesson observations indicated where and when difficulties arose.

In the Structured Group class difficulties on the computer assignments occurred when the level of the task was too advanced for the age group. Once the teacher modified the task to a simpler form, the students always experienced success. The attitudes and interest level of the students seemed to relate to the degree of frustration or success the students experienced on a task. When there was a good degree of success, the students were positive, perseverant and sometimes enthusiastic. When they were frustrated or unsuccessful, the students were more negative.

There were three students in the Structured Group who experienced more frustration and were not as successful as the other students during certain lessons. These students all required extra teacher assistance but they all experienced success by the end of the study. Their programming test results indicated that they had achieved some level of independent programming competency by the end of the eleven computer sessions.
In the Guided Discovery Group there was a high level of enthusiasm and excitement in the whole group throughout the eleven sessions. The students frequently sought the teacher's recognition of their projects. They called her to view their screens and they willingly shared their experiences at group time. For this group, the lesson observations usually documented a positive group attitude about the computers.

There were a few children who had difficulty learning through discovery. They appeared to have trouble taking a risk and taking the initiative to create something on their own without teacher direction and guidance. These children relied on their peers to give them direction. Once they had mastered a few commands and procedures, they practiced these on the computer and felt more comfortable about the experience. They did not reach the highest level of programming but their programming test results indicated that they did experience some degree of success. The teaching method used with the Structured Group may have been more appropriate for these particular children.

In both groups the attitudes of the students towards particular lessons was sometimes affected by hardware problems. For instance, pairing of students, because there were not enough computers for each child, often created a negative attitude. Students frequently preferred to work on their own. Another difficulty was that students (particularly in the Guided Discovery Group) preferred colour monitors and there was a mixture of black and white and colour monitors. Some students became quite upset when they were not assigned a colour monitor
which meant that they did not enjoy the session. Finally, the attitudes of the students during some lessons were affected by unforeseen difficulties such as the late arrival of the computers, disruptions (related to the school) in the lessons, and rescheduling of lessons.

The task analysis for the Programming Test suggests tasks which were easier and tasks which were more difficult for the students. The results indicated that the Guided Discovery Group had a higher degree of success on item one (enclosing a sticker on the screen) than the Structured Group. Students in the Structured Group may have benefited from more free exploration of moving the turtle around the screen. On item two (drawing an open rectangle), the Structured Group had a higher level of success. This may have been a result of the teacher continually reviewing the "Move" command with the Structured Group. This command enabled the turtle to move without leaving a line on the path it took. Knowledge of this command was necessary to be successful with this test item. The Guided Discovery Group was more successful on item three (drawing a trapezoid). During the lessons, drawing a shape with slanted sides was not a compulsory assignment for the Structured Group so these students may not have understood how to do this or may not have been able to figure out the procedure. There was little difference on the group results for item four (filling in the trapezoid with colour). About two thirds of both groups were successful with this task. The Structured Group was much more successful on item five (drawing a lollipop) than the Guided Discovery Group.
Students in the Structured Group were given a few lessons about drawing curves and circles. The teacher's assistance to the Structured Group in learning this procedure may have influenced their success rate on this task. Very few of the Guided Discovery Group attempted to draw circles during the lessons so they were probably unaware of the procedure. As a group the Guided Discovery Group performed very poorly on this task. For tasks such as drawing circles, where there is a definite trick or insight to the procedure, perhaps it is better to present the procedure to the students rather than wait for them to discover it. Results for both groups on the last item (drawing a sailboat), were similar and most students were successful with the task. Both groups had difficulty with certain items but they were successful with other items.

Conclusions - Teaching Technique and Learning Environment

The review of literature suggested various types of teaching techniques which involved different degrees of structure for teaching Logo. Papert (1980) insists that Logo should be taught using a discovery approach whereas many other researchers (Clements, Computers in Early..., 1985; Hawkins, 1985; Leron, 1985; Mullan, 1984; Vaidya, 1983) contend that some structure is necessary for introducing Logo or the children will not discover the "powerful ideas". As a result of these varying views, this
study used a structured teaching method for one group and a guided discovery method for the second group to introduce a single keystroke program.

The results of the study showed no significant differences between the Structured Group and the Guided Discovery Group on the Programming Test. The average scores on this test indicated that all students in the Structured Group made progress in learning to program with DELTA DRAWING. Consequently, using a structured approach seemed to be an appropriate method of teaching the single keystroke program, DELTA DRAWING. This group performed better on two tasks (items two and five) on the Programming Test than the Guided Discovery Group. Using the structured approach and teaching the specific technique needed to perform these tasks seemed to benefit the children since the mean score of the Structured Group was higher on these two tasks. As the researchers (Clements, Computers in Early... 1985; Hawkins, 1985; Leron, 1985; Mullan, 1984; Vaidya, 1983) suggest in the literature, sometimes structure and direction by the teacher is necessary. Since the Structured Group did not perform significantly better overall, a structured method cannot be recommended to be the sole method for introducing a single keystroke program.

In addition, the lesson observations indicated that the general attitude of the Structured Group varied much more radically than the Guided Discovery Group. The frustration level of this group was high at times possibly because of the project demands put on the students which were perhaps too advanced for
kindergarten children. It is important that lessons progress slowly making small steps from one concept to the next. The level of success on lesson tasks seemed to relate to interest level by the students. The higher degree of success on the task, the higher degree of interest and on-task behaviour. This study did not test this hypothesis but observation records suggested this outcome.

The Guided Discovery Group also made progress in learning to program with DELTA DRAWING so the guided discovery method also seems to be an appropriate method of teaching DELTA DRAWING. This group performed better on two tasks (items one and three) on the Programming Test than the Structured Group. This may have been due to the amount of time the Guided Discovery Group was given to play with the turtle and to explore and investigate the parameters of its movements. The Structured Group was not given an opportunity for this free exploration to determine the capabilities of the turtle. So, it also seems necessary, as Papert and his supporters (Dewitt, 1884; Enns, 1986; Maxwell, 1984) believe, to provide time for discovery and exploration when using a Logo-type program. The Guided Discovery Group did not perform significantly better than the Structured Group on the Programming Test so the guided discovery method cannot be recommended as the only method for teaching a single keystroke program either.

The lesson observations for the Guided Discovery Group indicated a high level of enthusiasm throughout the eleven lessons. This may have been due to the students being in total
control of their projects. They were proud of their discoveries and appeared delighted with their successes. They took ownership of their creations. Papert (1980) intended students to make discoveries on their own when using Logo and wanted children to be in control of their learning (Jaworski and Brummel, 1984) and feel a sense of power (Housey, in Jaworski and Brummel, 1984) over the computer. The observational records for the Guided Discovery Group indicated these positive features. Willis (1983) and Jaworski and Brummel (1984) have reported that a Logo environment also encourages sharing of ideas. In this study the teacher encouraged child-child interaction rather than teacher-child interaction at the computers as Collins and White (1984) suggested. Children in the Guided Discovery Group definitely shared their discoveries with each other and many children benefitted from this experience. It allowed students to incorporate commands and procedures into their drawings that they did not discover on their own and may not have discovered. The social interaction and the language (questioning and explaining) used was a very valuable experience. As well as working at the computers, the students also shared their ideas at group time. The group time also provided time for the teacher to give the students a minimal amount of direction and guidance. The important components of the Guided Discovery method seemed to include the free exploration time, the minimal teacher intervention, the social interaction and the group time. These should be incorporated into the teaching method used for introducing a single keystroke program.
The results of this study suggest that a combination of structured and guided discovery teaching methods would be most suitable for introducing a single keystroke program. The literature supports this conclusion. Leron (1985) refers to this type of environment as "quasi-Piagetian" where students are given some guidance but also time to think and explore independently. Clements (Computers in Early..., 1985) also believes that in the lessons there should be a balance of the teacher instructing the students and free exploration or student projects. Since neither group performed significantly better than the other group on the Programming Test, it is reasonable to conclude that it would be beneficial to the students to use a combination of both teaching methods. At times the teacher could provide time for free exploration, where teacher intervention is minimal so that students can make discoveries on their own. The teacher could also provide some structure to teach specific procedures that are not likely to be discovered by the students.

The learning environment encompasses the teaching method and also how the students work on the computers. In this study, the students worked both with and without partners. The student interviews revealed that a majority of students preferred to work on their own on the computer. The lesson observations reinforced this outcome. Students tended to work better on their own than with a partner. Partners often seemed to inhibit on-task behaviours. It was more difficult for children to concentrate and complete a task when they were with another child.
More students in the Guided Discovery Group preferred to work alone than in the Structured Group. The Structured Group was almost evenly divided between working alone and working with a partner. One factor affecting these results may have been that students in the Structured Group were assigned the same task whereas students in the Guided Discovery Group worked on their own projects. It would be easier to work with a partner on a common problem than to work with someone else on their project.

Social interaction, particularly with the Guided Discovery Group, still occurred when students were on their own computers. In this group when two students were on one computer they did not seem to work together on a particular project, but when they worked on their own computer they would share their ideas and help each other solve problems. The Structured Group sometimes worked together on a common problem but frequently each child waited for his turn to solve the problem independently. More disputes over computers were heard from the Structured Group but more of this group preferred to work with a partner. Most five and six-year-old children seem to be too self-centered to be able to solve a problem with someone else. A few children, however, preferred to have a partner when working on the computer because the other person helped them with difficulties. To conclude, sharing a computer at this age level does not seem to be advantageous for all children but may be suitable for some.
Conclusions - Benefits of Programming

The results of the study suggest that the computer programming experience with DELTA DRAWING had a positive effect on several aspects of the students' cognitive growth. Both groups made significant gains on the BRIGANCE Upper Case Letter Recognition Test, the BRIGANCE Directional / Positional Skills Test, the BRIGANCE Design Concepts Test and the Procedural Thinking Test. There were no significant gains, however, on the BRIGANCE Numeral Recognition Test. These results support the outcomes of previous studies. Programming with young children may improve performance on cognitive tasks but does not ensure performance improvement. This study, as in previous studies, shows both positive effects and no effects for cognitive growth.

There was no significant improvement on the BRIGANCE Numeral Recognition Test. The "ceiling effect" may have caused this outcome. Students were asked to identify the numerals from one to ten and the total possible score of the test was ten. Students in both groups scored well on the Pretest, and there was little room for improvement on the Posttest. For this study, it was not worthwhile testing beyond the numeral ten because the students would not be exposed to numerals higher than ten in the program. In a study conducted by Reimer (1985), who also used the BRIGANCE Numeral Recognition Test for numerals one to ten, he found no significant difference between Pretest and Posttest scores after a Logo programming experience on this test. Other
studies investigating the effects of programming on understanding mathematical concepts and skills have not specifically mentioned numeral recognition skills.

In this study students in both groups made significant gains from Pretest to Posttest on the BRIGANCE Upper Case Letter Recognition Test so the intervention of DELTA DRAWING affected the results on this test. In addition there was a significant difference between the Structured Group and the Guided Discovery Group with the Guided Discovery Group outperforming the Structured Group on both the Pretest and the Posttest. Since group differences were not apparent on any of the other tests, it is difficult to ascertain why the Guided Discovery Group achieved better results on this particular test. Reimer (1985) administered the same test but did not find any differences between the control group and the computer using group. Students in the Guided Discovery Group started out with a better knowledge of the alphabet. Students in the Structured Group made more of a gain between the Pretest and the Posttest than the Guided Discovery Group. It is difficult to conclude that the teaching method made a significant difference to the recognition of upper case letters even though there is a significant difference between the two group means. Further research is necessary to: 1) explore whether particular letters of the alphabet (related to the command keys) are better recognized after a computer programming experience; 2) explore whether oral discussion of the commands helps to improve letter recognition; and, 3) explore whether the teaching method affects the recognition of letters.
The conclusions from this study about the teaching method having an effect on letter recognition are very tentative and need to be supported in other studies.

The third test, the BRIGANCE Directional / Positional Test also resulted in differences between Pretest and Posttest scores. This study concludes that learning programming through a single keystroke program does improve students' understanding of certain directional/positional terms. This conclusion seems reasonable since DELTA DRAWING involves moving a turtle around the screen and putting it in different positions. Many of the commands, which the students learned, were directional/positional terms. Since both groups made significant improvements it is possible to conclude that the common experience of both groups, exposure to DELTA DRAWING, was the reason for better understanding of directional/positional terms. Reimer's (1985) study once again does not support these findings.

On the BRIGANCE Design Concepts Test the students also made significant improvements from Pretest to Posttest. Since both groups improved, the results suggest that learning a programming language can improve students' recognition of certain geometric shapes. The Edinburgh Logo Project (Howe et al., 1979) with 13-year-old boys and Reiber's (1985) study with grade two children found similar results and made similar conclusions.

Papert (1980) and his proponents (Clements, Computers in Early..., 1985; Kieren, 1984) believe learning to program in Logo promotes problem solving skills and the ability to think. This study investigated whether learning to program affects procedural
thinking skills in particular and concludes that computer programming indeed has a positive affect on procedural thinking skills. Both groups in the study demonstrated significant gains between Pretest and Posttest scores after the intervention of the single keystroke program. Students were better able to give directions, break a whole into its parts and sequence a set of pictures after the programming experience. During the "Giving Directions" section of the test a few students demonstrated that they had transferred the computer learning to off-computer tasks. They gave the teacher directions for moving by stating turtle commands (for example D,D,D and then R,R,R). This transfer may have occurred more naturally for the students in this study because they had used several off-computer supplementary activities during which it was common to instruct another child (the "turtle") how to move around.

Interestingly, on all of the tests administered there were no significant interaction effects between teaching technique and test level on the acquisition of specific skills. All significant effects were the result of the intervention, DELTA DRAWING or a result of the different teaching methods but not due to joint effects of both of these factors. A similar study over an extended period of time would be valuable to investigate these findings further.
Implications for the Classroom

This research study, which investigated the capabilities and benefits of kindergarten children to learn a programming language on the computer, has implications for the classroom. These will be discussed in this section.

The study concluded that learning to program is developmentally appropriate for kindergarten children. These students are capable of controlling the computer to have it perform a desired task. Consequently, students who use computers at the kindergarten level should be given the opportunity to work with a single keystroke program such as DELTA DRAWING. This experience can be positive and rewarding for them as well as challenging. The students take pride and delight in their accomplishments on the computer. It is appropriate to use single keystroke programs with young children because of their ease of use. Children can experience almost immediate success when using this type of program. These programs are a suitable means of introduction to Turtle graphics.

As a result of the conclusions of this study, several suggestions relating to the teaching method and general learning environment for introducing a single keystroke program can be made. A combination of both a structured teaching method and a guided discovery method is recommended. At times students need some direction and input from a knowledgable person (for example the teacher) to guide them on their computer activities but rigid structure is not always necessary and may stifle the students'
enthusiasm and excitement from making discoveries on their own. When direction and instruction is given it should be done so by taking small steps otherwise the students cannot make the connection between the two levels and they may become frustrated. It may be necessary (but not for all children) to provide instructions about specific commands or procedures to help them to make progress with their programming abilities. Otherwise, these students may stagnate because they do not or cannot continue to make discoveries which would bring them to a higher level of programming.

It is also important to provide considerable free exploration time when students are learning to program on the computer so that they can make discoveries on their own and chose their own projects. They seem to experience more pride over their own creations than when they have been directed to perform a specific task assigned by the teacher. They need the opportunity to explore, investigate and play with the program on their own with minimal teacher direction.

Social interaction between students should be encouraged when they learn to program on the computer. The teacher can direct the students to seek out knowledgeable peers to help with difficult situations or the teacher can suggest an "expert". Students should be permitted to move about freely to observe their peers and discuss problems with each other. They should also be encouraged to share their ideas during a group discussion period. These group periods are a valuable time for sharing knowledge and problems.
Teacher-child interaction should also occur in a setting promoting computer programming but the teacher should be more of a facilitator than an instructor. Guidance should be the teacher's main goal rather than instruction. Instruction will be a component of the program but not a major focus. Sometimes a structured lesson might be necessary to teach a specific procedure that is not likely to be discovered by the students. For certain children, depending on the learning style of the child, more direction and instruction may be necessary to avoid a negative attitude towards the computer and frustration by the student. The teacher should frequently make suggestions to the students but not always insist on a particular outcome. During group time the teacher can also share knowledge, sometimes through direct instruction and sometimes by asking guiding questions.

Students at the kindergarten level should work on the computers on their own initially unless they request to work with a partner. As the students gain competency with programming, solving problems with a partner may be more appropriate. It is important that the pairing does not allow one child to dominate the computer so teachers need to regulate these situations on the computer.

Teaching the students a programming language may have benefits for certain areas of cognitive development. Having students learn to program can be used by the teacher as another tool to assist students to make cognitive gains in various areas of the curriculum. The computer can be integrated into the
curriculum and programming introduced to provide another medium to promote cognitive development. The computer would not be used as the only means of teaching various skills (such as numeral recognition, upper case letter recognition, recognition of shapes, understanding directional / positional term) but it could be used to supplement and support other techniques. Creating a suitable teaching / learning environment, as described above, can also result in social development in the children. Since social skills are an important focus of a kindergarten curriculum, teaching programming for the social benefits that students could attain would be worthwhile in itself.

Recommendations for Future Research

Numerous recommendations for future research can be made as a result of this research study. Some of these recommendations are directly related to the conclusions. Others are not directly related but instead relate to issues about young children using computers and more specifically learning to program. This section will discuss both.

This study took place over a short period of time (one month). A future study should present a single keystroke program to kindergarten children much earlier in the school year and the study should continue for the whole year for a variety of purposes. First, the study could determine whether kindergarten students are capable of learning to program in the early months. It could also determine the capabilities of the kindergarten
students to program over a full year and discover their upper limits of programming, if any. Would any students reach the editing and debugging stage of programming in kindergarten?

The focus of another future study could be the effect of a quasi-Piagetian teaching style on learning to program. This style of teaching children to program would be a combination of a structured approach and a guided discovery approach. The programming capabilities of students at the end of the study could be explored. This type of study could also investigate how specific factors such as self-motivation, interest level of the child, intellectual level of the child, and motivation of the teacher all affect programming abilities of students.

DELTA DRAWING is just one single keystroke programming language. A comparison of different single keystroke programs and their capabilities may be worthwhile. It would be beneficial to determine whether students can make an easy transfer to Logo after using a single keystroke program i.e. was the single keystroke program a beneficial pre-Logo experience? It is also worth investigating whether certain single keystroke programs result in easier transfer to Logo than other single keystroke programs.

The difficulties experienced in each group of this study suggest that a particular teaching method may be better suited to certain children with a particular learning style. Investigating the interaction of teaching methods and learning styles when
children learn to program would be a worthwhile topic for a study. Students with a particular learning style may learn to program better if the appropriate teaching method is employed.

Further research needs to continue on the effects of programming on cognitive growth. Future research studies could focus on letter recognition and learning to program. One study could explore whether particular letters of the alphabet (related to the command keys) are better recognized after a computer programming experience. The study could also explore whether the teaching method affects letter recognition. A separate study could focus on directional / positional terms and could investigate whether students gain a better understanding of specific directional / positional terms which relate directly to the programming commands (for example right and left as opposed to over and under).

A study similar to the present study but over a longer period of time would be worthwhile to investigate whether there is an interaction effect between teaching technique and testing levels. The present study did not discover any joint effects.

The present research study also raised questions for future research in some related issues to young children and programming. One possible focus for a study could investigate relationships between the child's stage of development (in Piagetian terms) and the programming capabilities of the child. Is the child's programming abilities limited by their stage of development? Is development enhanced by programming?
Another topic for research could focus on the drawings produced on the computer. A researcher could investigate whether the level of drawings on the computer correlate with the level of drawings that the child does off the computer. Does the computer advance the child's ability to draw?

A final topic for research could compare DELTA DRAWING to Logo as a means of introducing Turtle graphics to kindergarten students. The programming capabilities of both groups could be compared.

This section suggests that much research needs to be done in the area of young children using computers and learning to program. The area is still in the developing stages. Unless conclusive research is made available, the computer in the Early Childhood area may be quickly dismissed as just another fad and its potential not truly realized.
REFERENCES


Appendix A. Procedural Thinking Tests

Procedural Thinking Test I

Part 1 - Giving Directions

1) Ask student to tell the teacher to go and touch the door.
   a) walk  b) touch the door

N.B. Teacher stands to start.

2) Ask student to tell the teacher how to paint a red line.
   a) pick up brush  b) put brush in paint  c) put paint on paper

3) Ask student to tell the teacher how to put a book away in a tote tray.
   a) walk to tote tray  b) pull out tray  c) put in book
   d) push in tray

4) Ask student to tell teacher how to go to the sandbox.
   a) walk to door  b) open door  c) walk through door
   d) turn right  e) walk

N.B. Teacher stands to start.

5) Ask student to tell teacher how to wash your hands.
   a) turn on water  b) take soap  c) wash hands  d) rinse with water
   e) turn off water  f) dry hands

N.B. Teacher stands at sink.

Part 2 - Breaking Whole Into Pieces

Teacher asks, "What do you need to ..."

1) Brush your teeth.
   a) toothbrush  b) toothpaste

2) Wrap this present.
   a) paper  b) tape  c) bow

3) Make a tomato and lettuce sandwich.
   a) bread  b) tomato  c) lettuce  d) knife

4) Set the table.
   a) tablecloth  b) fork  c) knife  d) plate  e) cup

5) Make a chinese lantern.
   a) large paper  b) strip for handle  c) crayons
   d) scissors  e) stickers  f) stapler
Part 3 - Sequencing

The students must put the sets of pictures in the right order.
1) Blow out candles on cake.
2) Dinosaur emerging from the water.
3) Building a snowman.
4) Making a bubble.
5) Planting seeds.

Procedural Thinking Test II

Part 1 - Giving Directions

1) Ask student to tell the teacher how to pick up lego.
   a) bend down   b) take lego block

2) Ask student to tell the teacher how to draw on the chalkboard.
   a) walk   b) pick up chalk   c) draw

3) Ask student to tell the teacher how to put a circle on a square (show example).
   a) cut out square   b) cut out circle   c) put glue on circle   d) put circle on square

4) Ask student to tell the teacher how to get a drink of water.
   a) turn on taps   b) pick up cup   c) put cup under tap   d) turn off water   e) drink water

5) Ask student to tell the teacher how to walk from the desk to the sink.
   a) walk   b) turn left   c) walk   d) turn right   e) walk   f) turn right

Part 2 - Breaking Whole Into Pieces

Teacher asks, "What do you need to..."

1) String beads.
   a) string   b) beads

2) Flowers in vase.
3) Cut hot dog.
   a) hot dog  b) bun  c) knife  d) mustard

4) House made of blocks.
   a) 1 large square  b) 2 small squares  c) large
       triangle  d) 6 small triangles  e) 1 rectangle

5) Blow paint picture.
   a) large white paper  b) paint  c) pencil
       d) straw  e) tissue paper  f) eye dropper

Part 3 - Sequencing

The students must put the sets of pictures in the right order.

1) Dinosaurs emerging from eggs.
2) Eating an ice cream.
3) Eating an apple.
4) Opening a present.
5) Diving into water.
Appendix B. Student Record Sheet -
Procedural Thinking Test

Name of child:

Part 1:

1) 1 2
2) 1 2 3
3) 1 2 3 4
4) 1 2 3 4 5
5) 1 2 3 4 5 6

Part 2:

6) 1 2
7) 1 2 3
8) 1 2 3 4
9) 1 2 3 4 5
10) 1 2 3 4 5 6

Part 3:

11) 1 2
12) 1 2 3
13) 1 2 3 4
14) 1 2 3 4 5
15) 1 2 3 4 5 6
Appendix C. Programming Test

1) Circle a sticker on the screen. 2 minutes

2) Draw the following shape on the screen. 2 minutes

3) Draw the following shape on the screen. 3 minutes

4) Fill in the shape in question 3 with colour. 2 minutes

5) Draw a lollipop. 3 minutes

6) Draw a sailboat. 5 minutes
Appendix D. Definition of Major and Minor Problems for Programming Test

Major Problems:

1. No resemblance to intended shape.
2. Unable to determine the correct keys.
3. Drawing not complete.
4. Not proper shape eg. straight-sided and it should be circular.
5. Sticker not enclosed (for item 1).
6. Major focus of drawing omitted.

Minor Problems:

1. Substituting shapes for parts of the picture eg. using a rectangle for the triangular sail on the boat.
2. Shape open and it should be closed.
3. Minor detail of shape not correct and all major parts correct.
4. Extra lines.
5. Part of drawing omitted eg. stick of lollipop.
6. Part of object or object turned incorrectly.
7. Incorrect openings. Too many or too few openings or openings not in the correct place (item 2).
8. Colour leaked out (item 4).
9. Filled wrong space with colour (item 4).
10. Stick omitted (item 5).
11. Stick too long (item 5).
12. Stick not straight (item 5).
13. Not correct placement of sail on boat (item 6).
Appendix E. Student Record Sheet - Scale for Programming Test

Name of child:

1 - No attempt

2 - Some attempt - major problems

3 - Good attempt - minor problems

4 - Success

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1)</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>2)</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>3)</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>4)</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>5)</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>6)</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>
Appendix F. Student Interview

1) Do you enjoy using the computers? Why or why not?

2) What is your favourite activity when we use the computers? Why is it your favourite activity?

3) What do you find hard to do on the computers? Why is this hard for you?

4) Did you like to use the computer to draw? Why or why not?

5) Which would you like to do more often - a) draw with a computer; or b) draw with a crayon and paper? Why?

6) How can the computer help you to draw?

7) What can you draw using a computer?

8) Can you draw anything on the computer that you cannot draw on paper? What is it?

9) Do you like pictures drawn using the computer? Why or why not? What do you like or dislike about the pictures?

10) How do computers draw pictures?

11) Do you like working alone or with a partner when you use the computer?
Appendix G. Parent Questionnaire

This questionnaire is a continuation of the study which you approved in February. I would appreciate your completion of this form. Please answer all relevant questions.

1) How many computers are in your home? ______
   If you do not have a computer in your home then go to question #16.

2) What type(s) of computer is it? ____________________________

3) Does your child use a computer in your home? Yes / No
   If no then go to question #16.

4) How long has your child used a computer?
   _____ Less than 6 months
   _____ 6 months to 1 year
   _____ 1 to 2 years
   _____ More than 2 years

5) How much time does your child spend on the computer per week?
   _____ Less than 1 hour
   _____ 1 to 2 hours
   _____ 2 to 4 hours
   _____ More than 4 hours. Specify amount__________________

6) Does your child work on the computer most days? Yes / No
   If no then how many days a week?__________________________

7) How much time does your child usually spend on the computer in one sitting?
   _____ Less than 15 minutes
   _____ 15 to 30 minutes
   _____ 30 to 60 minutes
   _____ More than 1 hour. Specify amount__________________
8) What types of activities does your child engage in on the computer (e.g., strategy games, educational games, programming)? Briefly describe each activity (e.g., name of the activity, what does the child do, what is the object of the activity).

1) ________________________________________________________________

2) ________________________________________________________________

3) ________________________________________________________________

9) Of these activities described in question #8, which is your child's favourite activity? Mark the appropriate box above with an X.

10) What does your child find difficult to do on the computer? Briefly explain.

______________________________________________________________

______________________________________________________________

______________________________________________________________

11) How does your child work on the computer?

____ Independently
____ With another child
____ With an adult
____ Other. Specify________________________________________

12) Has your child ever used a LOGO graphics program outside of the school? Yes / No
If no then go to question #17.
13) How long has your child used LOGO outside of the school?

___ Less than 1 month
___ 1 to 2 months
___ 2 to 6 months
___ More than 6 months

14) How does your child work with LOGO on the computer?

___ Independently
___ With another child
___ With an adult
___ Other. Specify

15) What does your child like to draw using the computer? Briefly describe.


* Question #16 is only to be answered if you DO NOT HAVE a computer in your home OR your child DOES NOT USE the computer in your home. Otherwise go to question #17.

16) Has your child ever used a computer? Yes / No

If yes then describe the occasion(s). Please indicate, if possible, the types of activities engaged in and the approximate duration of the session(s).


17) Do you support or oppose young children under eight years old using computers? Explain your answer.


Appendix H. Lesson Plans
Structured Environment
Lesson 1

Objectives:
1. To introduce the students to the computer program DELTA DRAWING.
2. To introduce students to the "turtle".
3. To have students emulate the turtle off the computer.
4. To introduce the students to the following commands: D, R, L, M and E.

Materials:
DELTA DRAWING disk
Apple IIe computers (or equivalent)
Turtle on a stick (multiple copies)
Wall charts of commands
Masking tape
Turtle Tracks* 1-c (see sample on page 206)
Turtle Tracks 1-oc (see sample on page 207)

Procedure:
1. Before the lesson begins use the masking tape to make a facsimile of a computer screen on the floor.
2. Introduce the program DELTA DRAWING to the children on the computer. Inform the students that they will be in control of the computer and will direct a "turtle" (the cursor) around the screen. The turtle will leave a trail behind it when it moves.
3. Introduce the commands D, R, L, M and E to the students using the charts for reference. Choose one child to be a "turtle". Demonstrate to the students how they can make the turtle move on the screen (made out of masking tape) using the five commands listed above. Allow the students to give the turtle directions.
4. Assign the children partners. Distribute the "turtles on a stick" to one child in the pair. Distribute the off-computer assignment, Turtle Tracks 1-oc, to the second child in the pair. Explain to the students that one child is the turtle and the other child reads the commands from the paper to direct the turtle to move. Direct the students to work on the project and to switch roles when the first child has given all the directions on the paper.
5. Gather the students together. Explain the computer assignment, Turtle Tracks 1-c. Students create lines on the screen in different positions. They also put the turtle in different positions. Assign the students partners and computers. Distribute the assignment.
6. The teacher should circulate around the computers and help students when difficulties arise.


Lesson 2

Objectives:

1. To review the commands - D, R, L, M and E with the students through activities both on and off the computer.
2. To have the students play turtle off the computer and focus on right and left turns.

Materials:

DELTA DRAWING disk  
Apple IIe computers (or equivalent)  
Story - *The Turtle and the Rabbit*  
Turtle Board Game  
Logo sticker pages  
Stickers  
Wall charts of commands

Procedure:

1. Read the story *The Turtle and the Rabbit* to the children. Emphasize that real turtles move slow and steady and so does the turtle on the screen.
2. Review the commands D, R, L, M and E and their meaning. Use the wall charts for assistance. Focus on the difference between D and M. D (Draw) moves the turtle and leaves a trail. M (Move) moves the turtle but does not leave a trail.
3. Play turtle off the computer focusing on turning the turtle left and right. Each child will put different coloured stickers on each hand to differentiate right and left. The teacher will direct all the students to move on the floor emphasizing the right and left commands.
4. Gather the students. Explain the computer task. The teacher puts a sticker anywhere on the screen. The students must point the turtle towards the sticker. When they finish, the teacher checks their screen and they put the sticker on their paper and get a new sticker on the screen (This is the sticker page system for the computers.). Assign the students partners and computers.
5. Explain the off-computer activity, The Turtle Board Game. Students put a board on the floor in front of them in a horizontal position and move small triangular cardboard turtles around it using the commands learnt. Assign students partners and distribute one board and one turtle to each pair. The teacher directs the students to move the turtle to the center of the board. Instruct the children to locate the nose of the turtle. Then, direct the turtles around the board.
6. Gather the materials and the group. Explain the second computer activity. This activity has the student move the turtle so that it hides under the sticker instead of pointing to it. The students tell the teacher when they have finished. The sticker goes on their sticker page and they get a new sticker on the screen. Assign the students partners and computers.

Lesson 3

Objectives:

1. To review the commands D, R, L, M and E with the students by manipulating the turtle through mazes both on and off the computer.

Materials:

DELTA DRAWING disk
Apple IIe computers (or equivalent)
Wall charts of commands
Masking tape
Maze transparencies
Logo sticker page
Stickers

Procedure:

1. Prior to the session make several mazes on the floor using the masking tape.
2. Review the commands D, R, L, M and E using the wall charts as references.
3. Gather the children and instruct them to sit beside one of the floor mazes. Choose one child to be the turtle and have individuals suggest how to move the turtle through the maze. Divide the class into four groups. Partner the children in each group. Have one child in each pair be the turtle and the other give the directions to the turtle about how to move through the maze. The children in each pair should switch roles. Then rotate the groups.
4. Introduce the maze transparencies. Demonstrate how to use them on the computer. Ask the children to direct the turtle through the maze and the teacher will type the commands on the computer.
5. Allow the students to choose a partner. Assign computers to each pair. The teacher attaches a maze transparency to each computer screen with masking tape. Instruct the children to complete the task and to show her when they finish. Students will receive a sticker for their sticker page upon successful completion. Then they will get a new maze.
Lesson 4

Objectives:

1. To review the commands introduced during the previous week.
2. To introduce the commands C, H, SHIFT G, ↑, ↓, → and ←.
3. To have the students draw curves on the computer.
4. To have the students draw letters on the computer.

Materials:

DELTA DRAWING disk
Apple IIe computers (or equivalent)
Wall charts of command
Masking tape

Procedure:

1. Prior to the lesson use the masking tape to form curves on the floor.
2. Review the old commands. Play a guessing game. For example, the teacher says, I'm thinking of a command which makes the turtle move without leaving a trail. What am I?" The children guess the proper command.
3. Introduce the new commands C, H, SHIFT G, ↑, ↓, → and ←. Use the wall charts to explain each command and the computer to demonstrate each command.
4. Discuss other types of lines the turtle can draw besides straight lines e.g. curves and shapes. Choose one child to be a turtle and have the students help the teacher direct the turtle around one of the floor curves. Repeat with another curve.
5. Discuss drawing letters on the screen using straight lines and curves. Assign partners and computers. Instruct the students to draw their initials on the screen using different coloured lines.
6. Continuing to work with partners, assign pairs to the floor curves. Have one student direct the other around the curve and then switch roles.
7. Reassign the students to the computers and instruct them to continue to try to draw letters.

Lesson 5

Objectives:

1. To review the commands introduced in Lesson 4.
2. To review how to draw a curve on the computer.
3. To have the students draw letters on the computer using letter transparencies.

Materials:

DELTA DRAWING disks
Apple IIe computers (or equivalent)
Letter transparencies
Masking tape
Turtle Board Game

Procedure:

1. Review the commands introduced in the previous lesson.
2. Discuss how to draw a curve on the computer. Elicit from the students the necessary commands. Direct all the students to find a space on the floor and pretend to be a turtle. The teacher instructs the students to walk out a curve on the floor.
3. Discuss the letter transparencies for the computer activity. Students move the turtle around the letter. When they finish, they press S, the turtle returns to the center and they get a new letter.
4. Assign half the students to the computers. Each child works on his own computer.
5. Assign the other half of the students to the off computer activity, The Turtle Board Game. The teacher directs the students to move the turtles around the boards to form curves, move in straight lines and make turns.
6. Switch the groups after fifteen minutes.
7. Allow the students to choose a partner and to work on the letter transparencies together on the computer.

Lesson 6

Objectives:

1. To review the commands introduced in the first week and the new commands introduced in Lesson 4.
2. To have the students draw curves on the computer.
3. To have the students draw letters on the screen using letter transparencies.
4. To introduce the Logo Game Board.

Materials:

DELTAR DRAWING disk
Apple IIe computers (or equivalent)
Wall charts of commands
Letter transparencies
Turtle Tracks 2-c--A to F (see sample on page 206)
Logo Game Board
Masking tape

Procedure:

1. Review all the commands introduced to date by playing a guessing game - What do I press if...?
2. Review the letter transparencies with the students. Do one with the whole group on the computer.
3. Introduce the new computer assignment, Turtle Tracks 2-c. Students type in the commands as given on the sheet and draw the picture on the paper that the turtle makes on the screen.

4. Assign half the group to computers (each student works on his own computer). Instruct students to do one letter transparency and one of the Turtle tracks 2-c (choose from sheets A to F). Switch the groups after fifteen minutes.

5. Assign the off computer group to the Logo Board Game. Four students work together at one board. The game reinforces the basic commands of DELTA DRAWING.

6. Gather the students. Direct the students to choose a partner to work on a computer. Students do either a letter transparency or a Turtle Tracks 2-c sheet.

Lesson 7

Objectives:

1. To have students identify the five basic shapes—circle, square, rectangle, triangle and diamond.
2. To have students determine the important features of the five basic shapes.
3. To introduce the shape transparencies for the computer.

Materials:

DELTA DRAWING disk
Apple IIe computers (or equivalent)
Attribute blocks
Pattern blocks
Turtle Tracks 2-oc (see sample on page 207)
Masking tape
Logo sticker pages
Stickers
Shape transparencies (see sample on page 205)

Procedure:

1. Prior to the lesson use the masking tape to form each of the five basic shapes—circle, square, rectangle, triangle and diamond on the floor.

2. Use the attribute blocks and the pattern blocks to have the students identify the five shapes and to discuss the important features of each shape.

3. Gather the students around one of the shapes on the floor. Discuss the shape. Choose a student to be the turtle and have the remaining students give the turtle instructions to move around the shape. Repeat this procedure with another shape.

4. Introduce the shape transparencies for the computer. Demonstrate on the computer. Attach the transparency to the computer with masking tape. Instruct the children to move the turtle around the shape.
5. Review the sticker page system. When students finish the shape transparency, they tell the teacher to receive a sticker and a new shape.

6. Introduce the off-computer paper project, Turtle Tracks 2-oc. Students copy the shapes drawn at the beginning of each line to fill up the line.

7. Divide the class into two groups. One group is assigned to the computers and works on the shape transparencies. The off-computer group has two projects. The children find a partner and work with that person to move around each shape on the floor. Each person has a turn being the turtle and giving the directions for each shape. When students complete this task they are given Turtle Tracks 2-oc.

8. Switch the two groups after fifteen minutes.

Lesson 8

Objectives:

1. To introduce the new commands U, CTRL F, CTRL N, CTRL E and to review S and C.
2. To have the students identify the five basic shapes.
3. To have the students follow directions to draw a shape on the computer.
4. To have the students make the four linear shapes (square, rectangle, triangle and diamond) on geoboards.

Materials:

DELTA DRAWING disk
Apple IIe computers (or equivalent)
Wall charts of charts
Geoboards
Elastic bands
Attribute blocks
Pattern blocks
Turtle Tracks 3-c-A to E (see sample on page 206)
Turtle Tracks 3-oc (see sample on page 207)
Logo sticker pages
Stickers
Shape direction wall charts

Procedure:

1. Introduce the new commands on the computer using the wall charts of the commands for reference.
2. Review each of the five shapes with the students. Use the attribute blocks and the pattern blocks to play a game. The teacher gives clues about each of the five shapes and asks the children to answer. For example, the teacher says, "I'm thinking of a shape with four sides that are all the same length. What am I?" The student must name the shape and find one in the set of blocks. Repeat for each of the shapes.
3. Introduce and discuss the shape direction wall charts.
4. Introduce the computer activity, Turtle Tracks 3-c, to the students. Students are given a shape direction sheet. Each sheet has directions for one of the five shapes in a small and large size. Students type in the directions to draw the shape on the computer. Students should cross off each direction when it has been typed. Then the shapes should be filled with colour. When the shape is completed, the student tells the teacher who gives him or her a sticker and another shape page.

5. Divide the class into two groups. One group is assigned to the computers to work on Turtle Tracks 3-c. The off-computer group is assigned two tasks. The first task is to follow the teacher's directions for Turtle Tracks 3-oc. Students colour the shapes on the paper the specified colour. The second task is to make each of the linear shapes on the geoboards using the elastic bands.

6. Switch the groups after fifteen minutes.

Lesson 9

Objectives:

1. To have students identify the five basic shapes.
2. To have students follow directions to draw shapes on the computer.
3. To have students identify the patterns in the directions for the shapes.

Materials:

DELTA DRAWING disk
Apple IIe computers (or equivalent)
Wall charts of commands
Shape direction wall charts
Geoboards
Elastic bands
Turtle Tracks 3-c (A to E)
Logo sticker pages
Stickers

Procedure:

1. Review the five basic shapes. The teacher holds up one of the five shapes and the students have to hop the correct number of times for each side e.g. three times for a triangle.
2. Review the computer project, Turtle Tracks 3-c, with the students. Demonstrate on the computer.
3. Assign the computer group to computers and direct them to try to complete two shape direction sheets (Turtle Tracks 3-c). When each shape is completed the students follow the normal procedure for the sticker system.
4. Distribute the geoboards and elastic bands to the off-computer group. Direct them to make each of the four linear shapes in different sizes on the geoboards.
5. Switch the two groups after fifteen minutes.
Lesson 10

Objectives:

1. To have the students recognize that they can draw any picture on the computer.
2. To have the students recognize that they can plan a picture on paper and pencil and then draw it on the computer.
3. To have the students recognize that they can plan a picture using blocks or geoboards and then draw it on the computer.
4. To introduce the students to breaking a complex picture into its components.
5. To expose the students to the potential of what they can draw on the computer.

Materials:

DELTA DRAWING disk
Apple IIe computers (or equivalent)
Experience chart paper
Marker
Paper and pencils
Geoboards
Elastic bands
Pattern blocks
Attribute blocks

Procedure:

1. Discuss the similarities and differences of using the computer and the turtle to draw a picture and paper and pencil to draw a picture.
2. Present four picture files from the DELTA DRAWING program disk e.g. Easter Egg, House, Sailboat and Starfish to allow students to view the vast potential of drawing on the computer.
3. Use the experience chart and the marker to discuss drawing a face. Focus on the parts of the face and the placement of each part. Have the students suggest how to draw the face on the paper. The teacher follows the students' directions.
4. Direct the students to draw a face on the computer and then to draw any picture. Record the students' ideas.
5. Assign half the students to the computers to work on the face and then their own picture.
6. Assign the other students to the geoboards, pattern blocks, attribute blocks or paper and pencil. The students choose which activity they would like to use. Instruct them to make a picture of something that they would like to draw on the computer.
7. Switch the two groups after fifteen minutes.
Lesson 11

Objectives:

1. To have the students draw any picture on the computer.
2. To have the students break the picture into its components.

Materials:

DELTA DRAWING disk
Apple IIe computers (or equivalent)
Experience chart paper
Markers
Geoboards
Elastic bands
Pattern blocks
Attribute blocks
Paper and pencils
Videotape machine
Videotape

Procedure:

1. Prior to the lesson set up the videotape equipment. Have someone operate the machine during the lesson.
2. Do an example of a picture that could be drawn on the computer on the experience chart e.g. house with trees. Have the students help determine the parts of the picture and where and how to draw them.
3. Ask the children what they would like to draw on the computer and record their thoughts.
4. Assign half the students to the computers to work on their own picture. When a student finishes a picture, save it on disk. Assign the other students to the off-computer activities. Students choose from the geoboards, pattern blocks, attribute blocks or paper and pencil to create a picture.
5. Switch the two groups after twenty minutes.

Guided Discovery Environment

Objectives:

The objectives for the eleven Guided Discovery lessons were:

1. To have the students explore DELTA DRAWING through free exploration.
2. To have the students choose from a variety of off-computer activities and to use these activities appropriately.

The materials and procedures for each of the eleven lessons differed and will be described below.
Lesson 1

Materials:

DELTA DRAWING disk
Apple IIe computers (or equivalent)
Turtle on a stick (multiple copies)
Wall charts of commands
Turtle Board Game

Procedure:

1. Introduce the program DELTA DRAWING to the children. Inform the students that they will be in control of the computer and will direct a turtle around the screen.
2. Introduce the commands D, R, L, M and E to the students using the wall charts for reference. Demonstrate each command on the computer.
3. Assign the children partners and computers.
4. Direct the students to explore the commands introduced in the lesson.
5. Off the computer direct the students to choose from the turtle on a stick activity or the Turtle Board Game. Explain to the students that the cardboard turtles in both activities represent the turtle on the computer screen and can be given commands. Direct the students to use the small turtles on the boards and the large turtles on the floor. Allow the students to explore these materials freely.
6. Gather the students to discuss computer and off-computer discoveries.

Lesson 2

Materials:

DELTA DRAWING disk
Apple IIe computers (or equivalent)
Turtle on a stick (multiple copies)
Wall charts of commands
Turtle Board Game

Procedure:

1. Discuss all the commands on the wall charts which were not introduced in Lesson 1. Use the computer to demonstrate the commands.
2. Assign partners to the students and computers.
3. Direct the students to explore all the commands on the wall charts.
4. Off the computer direct the students to choose from the turtle on a stick or the Turtle Board Game.
5. Gather the students for a discussion period.
Lesson 3

Materials:

DELTA DRAWING disk
Apple IIe computers (or equivalent)
Turtle on a stick (multiple copies)
Wall charts of commands
Turtle Board Game
Masking tape floor mazes

Procedure:

1. Discuss with the students the pictures which they have drawn on the computer. Guide the students to think of new kinds of drawings.
2. Instruct the students to choose a partner and assign the pairs to computers.
3. Direct the students to explore the DELTA DRAWING program.
4. Off the computer direct the students to choose from the turtle on a stick activity, the Turtle Board Game or the masking tape floor mazes (used with the Structured Group).
5. Direct the students to use the computers for a second session and then to choose from the off computer activities.
6. Gather the students to have a short discussion period.

Lesson 4

Materials:

DELTA DRAWING disk
Apple IIe computers (or equivalent)
Turtle on a stick (multiple copies)
Wall charts of commands
Turtle Board Game
Masking tape floor mazes

Procedure:

1. Lead a discussion period about the discoveries of the previous week's lessons.
2. Assign partners and computers.
3. Direct the students to explore the DELTA DRAWING program.
4. Off the computer ask the students to choose from the turtle on a stick activity, the Turtle Board Game or the floor mazes.
5. Gather the students for a sharing time.

Lesson 5

Materials:

DELTA DRAWING disk
Apple IIe computers (or equivalent)
Turtle on a stick (multiple copies)
Wall charts of commands
Turtle Board Game
Masking tape floor mazes - linear and curved

Procedure:

1. Seek the students' attention by reciting the poem "The Turtle in a Box". Teach the class the rhyme.
2. Guide the discussion period to include the following issues:
   a) explanation of what the wall charts are for and how they can be used.
   b) explanation of the difference between the commands D and M.
   c) can letters be drawn on the computer?
   d) what else can be drawn on the computer?
3. Divide the class into two groups. Direct one group to use the computers for free exploration of DELTA DRAWING. Direct the other group to choose from one of the activities. The choices are: turtle on a stick, the Turtle Board Game and the floor mazes (linear and curved). Review the rules of each activity.
4. Switch the computer and off-computer groups after fifteen minutes.
5. Assign the students partners and computers. Direct the students to use the computers for free exploration.

Lesson 6

Materials:

DELTA DRAWING disk
Apple IIe computers (or equivalent)
Turtle on a stick (multiple copies)
Wall charts of commands
Turtle Board Game
Masking tape floor mazes - linear and curved
Logo Board Game
Maze transparencies

Procedure:

1. Prior to the lesson, draw letters on a few of the screens to see if the students notice them.
2. During the discussion period review the rules of the off-computer activities and introduce the Logo Board Game (see Lesson 6 of the Structured Group for instructions).
3. Divide the class into two groups. Assign one group to the computers for free exploration. Make the maze transparencies (used with the Structured Group) available to this group. Direct
the off-computer group to choose an activity from the manipulatives—large and small cardboard turtles, the floor mazes or the Logo Board Game.

4. Switch the two groups after fifteen minutes.

5. Assign the students partners to work with for ten minutes on the computer. 6. Discuss the discoveries of the lesson during a sharing time.

Lesson 7

Materials:

DELTA DRAWING disk
Apple IIe computers (or equivalent)
Turtle on a stick (multiple copies)
Wall charts of commands
Turtle Board Game
Masking tape floor mazes - linear and curved
Logo Board Game
Maze transparencies
Shape transparencies

Procedure:

1. Prior to the lesson draw simple pictures, letters and shapes on a few of the computer screens.
2. Discuss the results of the previous weeks' lessons with the class.
3. Divide the class into two groups. Assign one group to the computers for free exploration. Make available the shape transparencies (used with the Structured Group). Off the computer direct the students to choose from one of the following activities: small and large cardboard turtles, floor mazes and the Logo Board Game.
4. Switch the two groups after fifteen minutes.

Lesson 8

Materials:

DELTA DRAWING disk
Apple IIe computers (or equivalent)
Turtle on a stick (multiple copies)
Wall charts of commands
Turtle Board Game
Logo Board Game
Geoboards
Pattern blocks
Attribute blocks

Procedure:

1. Discuss the results of the previous week's lessons.
2. Assign half the class to the computers for free exploration. Direct the remaining students to choose from the off-computer activities—small and large cardboard turtles, Logo Board Game, geoboards, pattern blocks, and attribute blocks.

3. Switch the two groups after fifteen minutes.
4. Gather the students for a discussion period.

Lesson 9

Materials:

DELTA DRAWING disk
Apple IIe computers (or equivalent)
Turtle on a stick (multiple copies)
Wall charts of commands
Turtle Board Game
Logo Board Game
Geoboards
Pattern blocks
Attribute blocks
Paper and pencils

Procedure:

1. Discuss the different types of pictures which can be made on the computer. Ask the students what they would like to draw during this lesson.
2. Assign one group to the computers for free exploration and the other group to the off-computer activities (choose from large and small cardboard turtles, Logo Board Game, geoboards, pattern blocks, attribute blocks, and paper and pencils).
3. Switch the groups after fifteen minutes.

Lesson 10

Materials:

DELTA DRAWING disk
Apple IIe computers (or equivalent)
Turtle on a stick (multiple copies)
Wall charts of commands
Turtle Board Game
Logo Board Game
Geoboards
Pattern blocks
Attribute blocks
Paper and pencils

Procedure:

1. Load a picture file from the DELTA DRAWING disk onto one of the computers.
2. Present the drawing to the students during group time. Load other drawings on the screen.

3. Assign students to either the computer group for free exploration or the off-computer group to choose from one of the following activities: large and small cardboard turtles, Logo Board Game, geoboards, pattern blocks, attribute blocks or paper and pencils.

4. Switch the two groups after fifteen minutes.

Lesson 11

Materials:

DELTA DRAWING disk
Apple IIe computers (or equivalent)
Turtle on a stick (multiple copies)
Wall charts of commands
Turtle Board Game
Logo Board Game
Geoboards
Pattern blocks
Attribute blocks
Paper and pencils
Videotape machine
Videotape

Procedure:

1. Before the lesson begins set up the videotape equipment.

2. During the group discussion have a number of students present their drawings from the previous lesson.

3. Divide the class into the two groups—computer and off-computer. The computer group engages in free exploration and the off-computer chooses one of the activities made available (large and small cardboard turtles, Logo Board Game, geoboards, pattern blocks, attribute blocks and paper and pencils)

4. Switch the groups after twenty minutes.
Sample Shape and Maze Transparencies
Sample On-Computer Turtle Tracks Assignments

1-c

Turtle Tracks

Draw these lines.

\[ \uparrow \downarrow \rightarrow \leftarrow \]

Put the turtle in this position.

\[ \rightarrow \vee \rightarrow \leftarrow \vee \vee \]

2-c

A Turtle Tracks

Type H
Type D D R R
Type D D D D
Type L L D D

Press \( \downarrow \leftarrow \uparrow \uparrow \)

Make a picture of the turtle tracks.

3-c

Turtle Tracks

Diamonds

<table>
<thead>
<tr>
<th>Type</th>
<th>H</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>HD</td>
<td>HR</td>
<td>HR</td>
</tr>
</tbody>
</table>

To make a small diamond.

<table>
<thead>
<tr>
<th>Type</th>
<th>H</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>HD</td>
<td>HD</td>
<td>HD</td>
</tr>
</tbody>
</table>

To make a large diamond.
Sample Off-Computer Turtle Tracks Assignments

<table>
<thead>
<tr>
<th>Turtle Tracks</th>
<th>Turtle Tracks</th>
<th>Turtle Tracks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. DDRDLD</td>
<td>Copy the shapes.</td>
<td>Follow the directions to colour the shapes.</td>
</tr>
<tr>
<td>2. LMMDD</td>
<td>□</td>
<td>□</td>
</tr>
<tr>
<td>3. DRMDLMD</td>
<td>△</td>
<td>△</td>
</tr>
<tr>
<td>4. RRDLDL</td>
<td>□</td>
<td>□</td>
</tr>
<tr>
<td>5. MLDMDLRD</td>
<td>□</td>
<td>□</td>
</tr>
<tr>
<td>6. DDRDDRDDD</td>
<td>□</td>
<td>□</td>
</tr>
</tbody>
</table>