ED 352 261 SE 053 313

AUTHOR Ahmed, Aqeel M.

TITLE Learning to Program and Its Transference to Students' Cognition.

PUB DATE 27 Aug 92

NOTE 99p.

PUB TYPE Information Analyses (070) -- Reports - General (140)

EDRS PRICE MF01/PC04 Plus Postage.

DESCRIPTORS *Cognitive Ability; Cognitive Development; Computers; *Computer Science Education; *Computer Uses in Education; Creativity; Educational Research; Elementary Secondary Education; Literature Reviews; Microcomputers; *Programing; Programing Languages; Science Education; *Thinking Skills

ABSTRACT

Teaching of computer programming in elementary and secondary schools has become a very popular activity. The purpose of this paper is to review and analyze the empirical research on the benefits of learning computer programming on students' cognition abilities. The paper focuses on the question: Do the skills of computer programming transfer to students' cognition? The paper is organized into three sections based on the measurement outcomes of each study. Section 1 contains seven studies that investigated the relationship of programming with the following general cognitive outcomes: metacognitive ability, cognitive development, cognitive growth, general ability, creativity, and achievement. Section 2 contains eight studies that analyze the relationship between learning computer programming and problem solving. Section 3 contains six studies that examine learning computer programming with the following specific cognitive skills: rule-learning tasks, development of thinking skills, reasoning skills, general variable skills, and planning skills. In the 21 studies reviewed one half showed learning computer programming did not affect student cognition, and the other half showed some positive effects. However, the author concludes that all of the studies were flawed. For example only one study reported the reliability and validity of its instruments. Based on the studies, the author concludes that there is no relationship between learning computer programming and student cognition. (PR)

LEARNING TO PROGRAM AND ITS TRANSFERENCE TO STUDENTS' COGNITION

Library Research Paper

Aqeel M. Ahmed
University of Bahrain
Computer Science Department

August 27, 1992

SE 053 313

TABLE OF CONTENTS

LEARNING TO PROGRAM AND ITS TRANSFERENCE TO STUDENTS' COGNITION

Introduction

In a large nationwide representative sample of schools, Becker (1983) found that 47% of the elementary schools and 76% of secondary schools now offer at least one course in computer programming. In fact, the teaching of computer programming in elementary and secondary schools has become a popular activity. And the often stated major goal of the course is to teach problem solving. Papert (1980) claims that programming environments such as Logo provide experience that may reduce time between a student's concrete and formal ability stages. From his prospective, our culture provides relatively little opportunity for systematic thinking that characterizes the stage of formal operations. Therefore, he suggests that the experience of planning, executing, and debugging programs in a computer-rich unstructured environment may help students to make the transition to formal systematic reasoning.

The literature is filled with claims that learning to program a computer provides for significant cognitive advances. Feuzeig, Horwitz, and Nickerson (1981) argued that learning programming can provide an opportunity to develop rigorous thinking, to learn to use a heuristic, to nourish self-consciousness about the process of problem solving, and in general achieve significant cognitive advances. Similarly, Linn (1985) analyzed the cognitive requirements of different levels of programming, such as precision and structural organization, in belief that these requirements transfer to process involved in general problem solving.

The purpose of this library paper is to review, analyze the empirical research on the benefits of learning computer programming on students' cognition abilities. The paper will focus on this question: Do the skills of computer programming transfer to students' cognition?

This paper is organized into three sections based on the measurement outcomes of each study. Section one contains studies that investigated the relationship of programming with general cognitive outcomes; seven of these studies are included in this section. Section two, contains studies that analyze the relationship between learning computer programming and problem solving; eight of these studies are also included in this section. Section three combines a variety of studies that examine learning computer programming with specific cognitive skills; six studies are analyzed in this section.

REVIEW OF THE LITERATURE

GENERAL COGNITIVE OUTCOMES

In this section seven studies are review and analyzed. These empirical studies investigate the effect of learning computer programming on general cognitive outcomes. General cognitive outcomes are as follows: metacognitive ability, cognitive development, cognitive growth, general ability, creativity, and achievement.

Clements and Gullo (1984) compared the effects of learning Logo programming with learning Computer Assisted Instruction (CAI) with respect to students' cognitive style, metacognitive abilities, cognitive development, and the ability to describe directions. The subjects were 18 first grade students from a Midwestern middle school class. The average age was 6 years and 11 months. The study had two groups, the CAI group and the Logo programming group. Subjects were randomly assigned to either a Logo group or CAI group.

All subjects were given a pretest on receptive vocabulary and reflective and divergent thinking, using the Matching Familiar Figures Test (MFFT) and the Peabody Picture Vocabulary Test (PPVT). The PPVT test was implemented to assess the receptive vocabulary (internal consistency reliability=.77). MFFT was used to measure reflectivity where the child was shown a picture of familiar objects, together with an array of similar variants.

Only one picture in the second array was similar to one picture in the first array. Here the child was asked to find the similar picture. The test-retest reliability ranged from .58 to .96. The Torrance Test of Creative Thinking Figural was administered to measure the ability of the child to think divergently in a nonverbal mode. The test was used to assess the child's ability in four manners: fluency (the number of original ideas the child had; test-retest reliability=.7), flexibility (the variation of idea for other idea; test-retest reliability=.83), originality (how original ideas compared to normative group; test-retest reliability=.85); and elaboration (how much detail added to main idea; test-retest reliability=.83). Markman (1977) designed two tasks to assess children's ability to monitor and evaluate their own cognitive process (metacognition). In this measure, children were given instruction on how to perform two tasks, but these instructions were incomplete. The first task contained eight alphabet cards that were divided between the child and the experimenter. In this task, each player laid out a card at a time, looking for a special card; however, there were no instructions on what the special card might be. The second task asked the child a similar magic trick. Markman's (1977) criteria for relevancy was used. The interrator agreement was 95%, and the test-retest reliability=.73. To measure the operational competence (classifications and seriation), four tasks were presented. Two classification tasks asked subjects to sort and resort geometric shapes and familiar objects (internal consistency reliability=.75). Two seriation tasks asked the subjects to order a series of objects by length (internal consistency reliability=.81). The McCarthy Screening Test (MST) was used to measure the cognitive development. This measure had four subsets: right to left orientation (internal consistency reliability=.32), verbal memory (internal consistency reliability=.54), draw a design (internal consistency reliability=.67), and numerical memory (internal consistency reliability=.69). A map test was also used to measure the ability to describe directions. Subjects were given a street map and asked to draw a path from their house to a certain store; then they were asked to describe it. Subjects were evaluated on their drawing of the path and describing its direction. The test-retest reliability was .91.

Subjects were given a sequence of learning sessions either in Logo programming or CAI. The CAI sessions consisted of computer-based lessons that concentrated on reading and arithmetic concepts. The Logo session consisted of a sequenced lesson in Logo programming.

The study lasted for 12 weeks, and two 40-minute sessions were presented per week. Researchers worked in rotation with the CAI and the Logo groups. At the end of the 12 weeks, subjects were given a posttest on the following concepts: cognitive style (reflective and divergent thinking), metacognitive, cognitive development (operational competence and general cognitive measures), and the ability to describe direction.

Metacognitive refers to a knowledge about and monitoring of ours own cognitive processes such as thinking, learning, and remembering. Metacognitive ability refers to the ability to recognize when one strategy is not working and a new strategy is called for (Woolfolk,1987).

The t-test results revealed no significant differences between Logo and CAI groups on their PPVT scores. The analysis of variance was performed between pretest and posttest scores on the Torrance test for CAI and Logo groups. The results of the analysis revealed significant main effects for the group, F(1,48)=7.44,$p<.05$; pre-post, F(1,48)=8.83,$p>.01$, and subtest, F(1,48)=19.36,$p<.01$. In order to determine where the significant differences occurred in the interaction, post hoc analysis (Scheffe) was performed. In the Logo group there were significant differences from pretest to posttest on the following subtests: fluency, $p<.01$ (means for pre and posttest were 46.00, 58.89, respectively); originality, $p<.05$(means for pre and posttest were 51.67, 66.56, respectively); and overall divergent thinking score, $p<.01$ (means for pre and post were 168.56, 228.56 respectively). Correlated t-tests were also performed between the pretest and posttest scores on the MFFT for the Logo and CAI groups. The analyses revealed a significant difference between pretest and posttest scores for errors, t(8)=3.58,$p<.007$, and latency, t(8)=2.21,$p<.05$. The means for the Logo group in error and latency were 16.56 and 6.94 respectively. The means for the CAI group in latency and error was 14.11, 10.11 respectively.

The Logo and CAI group scores on the remaining measures were analyzed using the t-test. The analysis showed that the Logo group needed significantly fewer probe questions on the two metacognitive tasks than the CAI group: game task t(16)=3.54,p<.01; magic task, t(16)=-2.87,p<.02. The means for Logo in game and task were 4.33 and 1.78, respectively. The mean for CAI in game and magic were 9.11 and 6.11, respectively. On the describing directions test, the Logo group scored significantly higher than the CAI group t(16)=4.42,p<.001. The mean for the Logo group was 8.8, and the mean for the CAI group was 7.7. The comparison of the results of the remaining measures between Logo and CAI showed no significant differences.

The researchers concluded that Logo programming achieved better than the CAI group on measures of metacognitive ability and ability to describe direction. The researcher also found no difference on measures of cognitive development.

The study did not mention how the subjects of the sample were selected, nor the socio-economic status of the subjects. There was no validity reported for any instruments used in this study. Moreover, the MST had low reliability in all parts respectively. The reliability of MFFT was low (.58 to .96), and also more details were needed on the PPVT measure.

For analyzing the data collected by both instruments, MFFT and Torrance, the ANCOVA instead of the ANOVA or correlated t-test needed to be implemented so that the posttest means can be adjusted to compensate for pretest differences.

Clements (1986) conducted a study to assess the effects of learning computer programming and computer-assisted instruction (CAI) on cognitive skills (classifications and seriation operation), metacognitive skills, creativity, and achievement (reading, mathematics, and the ability to describe directions). Seventy-two children from a middle-class school were randomly selected. There were 36 first grade (18 girls and 18 boys, mean age=6 years, 10 months) and 36 third grade children (19 girls and 17 boys, mean age=8 years, 10 months). All subjects were randomly assigned to one of the three treatments: CAI, Logo programming, or control.

All subjects were given pretests on operational competence (classification and seriation), creativity, reading, and mathematics achievement. Operational competence (classification and seriation) contained four tasks. Two classification tasks asked subjects to sort and resort geometric shapes and familiar objects (internal consistency reliability=.75). Two seriation tasks asked the subjects to order a series of objects by length (internal consistency reliability=.81).

The assessment of metacognitive skills consisted of an eight-item test to measure the metacomponents. Metacomponents are executive processes that are used in planning and evaluating one's information processing. Before the test, subjects were shown a 3 (shape) X 3 (color) array of geometric pieces with two pieces missing. Then students were asked to find the correct missing pieces. The other metacomponent questions were similar to the above question (internal consistency reliability=.81). The Matching Familiar Figures Test was then implemented to measure cognitive skills. This measure is described in previous study (Clement and Gullo, 1984). Markman (1977) designed two tasks to assess children's ability to monitor and evaluate their own cognitive processes (metacognition). This measure is explained in previous study (Clement and Gullo, 1984).

Metacognitive abilities, metacommunicative, metamemorial, and metasocial-cognitive tasks were administered to the subjects. Metacommunication is a hidden message of a communication. Although this may be not the message the speaker intended to send, it may be the message that it received (Woolflolk, 1987). Metamemorial is awareness of memorization or beyond memorization. Metasocial-cognitive is awareness of learning from socializing or cooperative learning.

The first metacommunication task required children to listen to a story. This story was about a boy providing directions to another boy. The students were asked to indicate how the first boy knew that the other boy didn't understand the directions, which of the two sets of directions to the house is superior, and to justify their choice. Here the metamemorial tasks were to measure the awareness of memorial strategies. The metasocial-cognitive tasks asked children how they could tell what the other child was thinking about (Interrater agreement=94%).

The Torrance Test of Creative Thinking Figural was administered to measure the ability of the child to think divergently in a nonverbal mode. This instrument is described in the previous study (Clement and Gullo, 1984).

Subjects were given a street map and asked to draw a path from their house to a certain store; then they were asked to describe it. This assessment is described in the previous study (Clement and Gullo, 1984).

At the beginning of the year students were given achievement tests; these achievement tests measured students on reading and mathematic skills. The spilt half reliability for reading and mathematic skills were .94 and .80, respectively. The study lasted for 22 weeks where a sequence of 44 sessions in either Logo (Terrapin) programming or CAI was presented. During each session, students worked on the computers under the guidance of one or two teachers (a graduate-assistant experienced in teaching Logo programming and a researcher).

In the Logo programming treatment, children were given a sequence of lessons. In Logo students concentrated in learning Logo programming. In the CAI treatments, students were given a sequence of lessons using commercial computer programs. These commercial computer programs concentrated on teaching drills, tutorial, problem solving, arithmetic, and reading. The control group simply participated in regular schedule classroom lessons. At the end of the study all subjects were given a postest to assess their cognitive skills, metacognitive skills, creativity, and achievement.

Analysis of variance 3 X 2 X 2 (Group X Grade X Pre-Post) was performed to determine if there were significant differences among scores on the classification tests. Analysis showed significant main effects for grades, F(1,63)=18.13,p<.001; and pre-post, F(1,63)=24.42,p<.001; and significant interactions for group X pre-post, F(2,62)=4.6,p<.05; and grade X pre-post, F(1,63)=4,86,p<.05.

The results of the Newman-Keuls tests showed that the Logo posttest scores for third grade students were significantly higher than all other scores such as, third grade pretest, first grade pretest, and posttest (p<.01). For Group X Pre-Post, the Logo posttest scores were significantly higher than CAI posttest (p<.05), CAI pretest (p<.01), and the Logo pretest (p<.01) scores. The CAI posttest score was significantly higher than the CAI pretest and the Logo pretest scores (p<.05). The control posttest score was significantly higher than the CAI posttest score (p<.05) and the Logo and CAI. The control pretest scores were significantly higher than the Logo and CAI pretest scores (p<.05).

A 3 X 2 X 2 (Group X Grade X Pre-Post) ANOVA was performed on the seriation test. The analyses showed a significant main effect for grade, F(1,63)=10.75,p<.01; and pre-post, F(1,63)=24.56,p<.001; and significant interactions for Group X Pre-Post, F(2,63)=6.8,p<.01 and Group X Pre-Post, F(2,63)=5.04,p<.01.

An analysis of variance 3 X 2 X 2 (Group X Grade X Subtest) was performed to determine if there were significant differences among scores on the test deciding on the problem and on solution processes (the first four metacomponents). The analysis revealed significant main effects for group, F(2,63)=6.85,p<.01; and grade, F(1,63)=7.10,p<.01; and significant interactions for Grade X Subtest, F(3,189)=6.09,p<.001. The Logo group outperformed both the CAI and the control group (p<.01), and the third graders scored higher than first graders.

The analyses for the MFFT reflective index showed only the main effect for the grade was significant, F(1,63)=11.03,p<.01, and the third grade scores were higher. An ANOVA performed on the score for comprehension monitoring showed the main effects for group, F(2,61)=6.61,p<.01; and grade, F(1,61)=3.39,P<.01. The comparisons between Logo and CAI indicated that the Logo group scored significantly higher than both the CAI and the control groups (p<.01). The third grade score was higher than the first grade score.

An ANOVA performed on the metacognitive measures from other domains revealed a significant main effect for grade, F(1,61)=22.88,p<.001, with the third grade scoring higher. A 3 X 2 X 2 X 4 (Group X Grade X Pre-Post X Subtest) ANOVA was performed to determine if there were significant differences among scores on the Torrance test. Analyses revealed a significant main effect for group, F(2,64)=11.16.p<.001; grade, F(1,64)=69.37,p<.001; pre-post, F(1,64)=38.23,p<.001; and subtest, F(3,192)=10.76,p<.001; and significant interactions for group X subtest, F(3,193)=55.92,p<.001; and group X pre-post X subtest, F(6,192)=5.53,p<.001.

An analysis of covariance was performed with the pretest as the covariate, because different tests were used to measure pretest and posttest achievements. The analyses showed that there were no significant effects for reading or mathematics achievements. A 3 X 3 (Group X Grade) ANOVA was performed on the directions tasks; this revealed main effects for group, F(2,63)=20.64,p<.002; and grade, F(1,63)=7.4,p<.01. Comparisons between groups showed that the Logo group out-performed the control and the CAI groups, (p<.01) and that third graders outperformed first graders.

The researcher concluded that the Logo group outperformed the CAI and control groups on measures of operational competence. Moreover, the researcher found no differences on the measure of reading and mathematic achievements.

The study did not provide enough information on how the sample was selected. The operational competence instrument was implemented to measure the children's abilities in logical operations of classification and seriation. However, the researcher failed to report the validity of this instrument. The researcher developed an eight-item test to measure the metacomponents; the researcher, however, failed to report the validity of this measure. Also, the reliability for this measure was low. Furthermore, MFFT was used to assess the metacomponets of allocating time resources. The reliability of MFFT was also low. The study did not report the validity of this instrument.

A measure developed by Markman was implemented to measure the ability to monitor cognitive processes. The study failed to report the validity of this measure.

Metacognitive abilities, metacommunicative, metamemorial, and meta-social-cognitive tasks were administered to the students; however, the researcher failed to report the validity of each task. Also, the study did not mention how the interrater agreement was established.

The Torrance test of creative thinking assessment was used to measure the subjects creativity. Here again, the researcher did not mention the validity of this instrument.

An achievement test was implemented to measure the students' skills in reading and mathematics; however, the researcher failed to report the validity. The researcher developed a map test to determine children's ability to describe directions. Here again, the researcher failed to mention the validity of this assessment.

The analyses of ANOVA on the seriation test showed a significant main effect for grade. However, the study did not mention where it was significant. Thus post-hoc analysis should be performed to find where it is significant. Also post-hoc analysis needed to be applied for the cognitive measure results to find where it was significant.

The study mentioned that the Logo group outperformed both the CAI and the control groups, and the third grade of Logo group scored higher than the first grader of the same group; however, the researcher fail to report the statistical method and values to support these results.

It appeared from this study that different teachers taught the Logo and the CAI groups. There was a possibility of bias because of the different teachers and it would be preferable if these teachers were rotated in all three groups in order to avoid bias in teaching. Also, the study did not provide enough information about what the control group did study. In addition, ANCOVA instead of ANOVA should be performed for analyzing the data.

Turner and Land (1986) investigated the effects of learning Logo on understanding specific mathematical concepts and the level of cognitive development. The following are the hypotheses of this study: (1) when students learn Logo they will achieve higher on mathematics concepts and attain a higher level of cognition than those who do not learn Logo; (2) students who learn more in Logo will achieve significantly higher on mathematics concepts and attain a higher level of cognition than those who learn a minimal amount of Logo.

The sample consisted of 181 subjects from seven classes in four inner-city schools in the Midwest. The experimental group, which was called the Logo group, consisted of 91 students who studied Logo for one hour a week for 16 weeks as part of their mathematics curriculum. The goal of Logo was to supplement the learning of these specific mathematics concepts: estimation of angles and distance, properties of polygons, variables, rectangular coordinate system, negative numbers, perimeter, and symmetry. The control group consisted of 90 students who studied mathematics for 16 weeks. Fifty-nine fifth and sixth grade students from a magnet school participated in one of the experimental groups. These students scored in the top three stanines on standardized reading and achievement tests. The other experimental group consisted of 32 students who were 6th-8th graders from another school, and these students scored in the lower six stanines. Students in the two control groups were 6th-8th graders and represented all the ability levels.

All students were pretested on selected mathematics concepts and cognitive developments. Students in the Logo group were given a posttest on Logo programming. A Social Science Piagetian Inventory (SSPI) was implemented to measure cognitive development. This instrument contains 30 multiple choice questions. The authors of this study indicated that this measure had concurrent validity determined by administrating both the SPPI and individual clinical interviews to 30 students. The students' ages ranged from 10 to 13 years; the classification agreed 60 percent of the time. The SSPI had a reliability of .87. In measuring the students on the mathematics concepts, the researchers developed a 22 multiple choice question test. This test covers the following mathematic concepts: angles, estimation of distance and area, properties for polygon, rectangular coordinate system, variable, negative numbers, symmetry, and perimeter. The reliability coefficient of this test was .87 for 141 students in grade 5-9. The Logo test was developed by the researchers; this test contains 20 multiple choice questions. The researchers indicated that the Logo test had content validity because its items were developed from the specific objectives of the Logo activities. The spilt-half reliability coefficient was .70, based on a sample of 51 middle school students.

In comparing the means and standard deviations for the Logo group and the control group on each variable, both pre-math and Pre-SSPI, the means score for the Logo group were significantly higher than the mean scores of the control group prior to the treatment (t=3.4,p<.01 for math-pre, and t=3.5,p<.01 for SSPI-pre). The means for the Logo group on the pre-math and pre-SSPI were 11.30 and 14.12 respectively. The means for the control group on the math-pre and pre-SSPI were 8.63 and 11.54, respectively.

The study found that there were significant correlation coefficients between the math-post and math-pre (r=.76,p<.001), SSPI-pre and math-pre (r=.71,p<.001), SSPI-pre and math-post (r=.66,p<.001), SSPI-post and math-post (r=.66,p<.001), SSPI-post and math-pre (r=.72,p<.001), SSPI-post and math-post (r=.75,p<.001), SSPI-post and SSPI-pre(r=.72,p<.001), Logo and math-pre (r=.63,p<.001); Logo and math-post (r=.65,p<.001), Logo and SSPI-pre (r=.49,p<.001), and Logo and SSPI-post (r=.63,p<.001). The results of one-way analyses of covariance of math-post and SSPI-post by treatment group, using the appropriate pretests as covariates, showed that there was no significant difference between the Logo group and the control group in mathematics achievement. Also, no differences existed between the Logo and the control groups in the level of cognitive development.

A criterion group design was applied to investigate the differences between those who learned minimal and more amounts of Logo. The students who scored more than one-third of the standard deviation above the mean in the Logo group were called the high Logo, and those who scored one-third below the means were called the low Logo.

One-way ANCOVA of Math-post and SSPI-post by Logo level, using the appropriate pretests as covariates, were performed. Analysis revealed that there were significant differences in favor of the high Logo group in their understanding of mathematic concepts and also in their cognitive development (F(1,49)=5.51,p<.02 and F(1,41)=8.50,p<.01), respectively).

The researchers concluded that learning Logo programming does not effect students' cognitive development nor their achievement in mathematics. However, there was a significant difference between students learning only a minimal amount of Logo and students learning most Logo in favor of high Logo group. These findings suggest that cognitive development, achievement in mathematics, and achievement in Logo programming all share a common factor, that students who do well in one area are likely to do well in other areas.

There is a possibility of regression toward the means in the study because in the experimental group there were two schools in which students in one school scored at the top three stanines on standardized test (mathematics and reading) while the students in the other school scored in the lower six stanines. The study indicated that Carter and Ormord provided evidence of the concurrent validity of the test by administering both the SSPI and individual clinical interviews to 30 students ages 10 through 13, and the classification agreed 60 percent of the time. The study did not provide information about these 30 students such as their socio-economic status or if these students came from a population similar to the population of this study. Furthermore, the classification agreed only on 60 percent which was too low to provide a solid evidence of concurrent validity. Thus, the reported concurrent validity was not strong enough to consider SSPI a valid instrument. The students in the test-retest reliability were from grades 5-9 while the students in this study were from grades 5-8. Again, there was an age difference. The researchers did not report validity and reliability for the math test that was used to measure the students' abilities in mathematics.

The researchers developed a 22-question multiple choice test covering several mathematical concepts. The reliability coefficient of this test was .87 for 141 students in grades 5-9. The researchers did not mention if these 141 students came from a population similar to the subjects in this study.

The study considered the unit of analysis as the number of the subjects which was 126. Here seven classes were selected; therefore, the group is the unit of analysis, six rather 126. The study did not mention the number of teachers who participated. It also was not clear whether there was one teacher or two for the experimental group and the control group. Therefore, there might be a teacher effect.

Researchers found that there was a significant correlation coefficient among all measurements, (r=.76). This correlation coefficient was not of practical value because r^2=.5776 is low. This applied for all other significant correlation coefficients because these correlation coefficients were low and considered not of practical value.

Linn and Dalbey (1985) studied the cognitive consequences of programming instruction and assessed the following: general ability, access to computers outside the school, previous computer experience, and interest shown in computers. The language used for the instruction in this study was BASIC. Sites within 50 miles of the Lawrence Hall of Science, the University of California, in Berkeley, were surveyed for selection. Schools offered rigorous computer programming courses. The rigorous programming course was to be at least 12 weeks in duration, and to have at least eight computers available for students and teachers with at least 100 hours of experience in programming. For Sites 1-3, three schools that were selected used Apple computers. Site 4, a school, was selected close to the Lawrence

Hall of Science. At Sites 5 and 6, schools with exemplary instruction were selected. The Advanced Progressive Matrices, Set I (Raven 1965) was used to measure the ability of the subjects. This test has 12 items which had alpha reliability of .77. The students were divided into low, medium, and high ability groups. The low ability group scored between 0 and 6 on the Raven. These students had difficulty in understanding the matrices. The medium ability group scored between 7 and 10 on the Raven, these subjects understood the principles of the matrix item. The high ability group consisted of those students who scored 11 or 12; these students fully understood the matrix.

A test called Headlines was implemented to assess interest in the following: computers, science, and non-technical areas. The test containing 24 items, consisted of 12 on computers, eight items on science, and four non-technical items. The alpha reliability for the total test, computer items, science items, and nontechnical items were .83, .78, .78, and .56, respectively. The consequence of cognitive measured by the Final Programming Assessment test. This test contains three sections: comprehension, reformulation, and design. In the comprehension part students were asked to predict the output of programs. The alpha reliability for this section was .93. In the reformulation part students were asked to modify programs, and the alpha reliability for this section was .81. In the design segment students were asked to write programs and the alpha reliability for this section was .91. The whole test had an alpha reliability of .96. The authors indicated that this test had face validity. Participating teachers reviewed the content and agreed that the test was measuring what they taught.

The results of the Final Programming Assessment at Sites 1 through 4 showed that there was diversity among students. For the comprehension section the means ranged between 20% and 70% correct. For the reformulation section, the means ranged between 15% to 50% correct. For the design section, the means ranged between 10% and 30% correct.

At Sites 5 and 6 (exemplary sites), the students mastered comprehension (mean scores between 78% and 95%). In the reformulation sections the mean scores were between 60% and 80% correct. In the design section students averaged 60% of items solved correctly at Site 5, while 80% were solved correctly at Site 6. At these two sites students developed more cognitive consequences than students at other sites.

The correlation coefficient between the performance for all students in all sites was .56. Significant difference between the means for the ability groups and programming performance at typical sites were found. At exemplary sites, there were no differences between groups of medium and high abilities. There was a significant relationship between both in-school and out-of-school access and performance.

The finding of this study revealed that the form of instruction, the access to computers, and the students' ability influenced outcomes from programming instruction. The researchers found that exemplary instruction moves students further along the chain of cognitive accomplishments than does typical instructions.

The sample consisted of six sites. In Site 1, there were 92 students from grade 8, only eight computers in that school available for the students' use, and the access time for the students to use the computer was 25 hours. In Site 2, there were 84 students from grades 7 and 8 with only 12 computers. The access time for these students was 26 hours. In Site 3, there were 132 students and only eight computers with the access time of 13 hours for each student. Obviously the access time for each student at each site was different. When schools have more computers and fewer students, students will have more access time. Therefore, it was more likely that students who had more access time will achieve better in programming than students who have less access time. Students should be given the same amount of access time in use of computers.

The Advanced Progressive Matrix, Set I, was used to measure the ability of the students; it was unclear what type of student ability this test measured. Also, the validity of this instrument was not reported.

The test called Headlines was used to assess interest in computers, science, and nontechnical areas. Reliability for the nontechnical items was low because this subtest had a few items and therefore more items in needed to increase the reliability. Also, the validity for this instrument was not mentioned. The final programming assessment was used to measure progress along the chain of cognitive accomplishments. This test was viewed as having face validity for the programming contents of courses involved in the study. The face validity was not enough. Content and construct validity were needed to support the validity of this instrument.

The results from the performance of the Final Programming Assessment showed that student outcomes were inconsistent. For instance, in the comprehension section the standard deviation was greater than the means at Site 3 and close to the means at Site 2. Furthermore, the standard deviation in the design section was greater than the means for all groups and this show that the distribution was less peaked.

The correlation between the programming assessment for all students in the sample was .56. The study did not indicate whether the correlation was significant. In fact this correlation was low, and thus it was not of practical value because the r^2 was low.

This study indicated that a significant difference existed between the means for the ability group and the programming performance at typical sites. However, there were no statistical values, unit of analysis, and alpha that supported these results. Also, the study stated that there was a significant positive relationship between both in-school and out-of-school access and performance. Again, no statistical value, unit of analysis, and alpha supported these results.

Howell, Patrick and Diamond (1987) studied the effect of Logo on the cognitive development of 5 to 7-year-old children. Their study was based on this question: Does learning Logo accelerate cognitive development? Cognitive development was measured by using the Piagetian tests. These tests were administered on the pretest and postest and measured the conservation of length, measurement, and the ability to identify Euclidean shapes. Instant Logo was used in the study as the programming language. The study lasted for six months with an individual session of 15-20 minutes per day.

The independent variable was learning Logo language, while the dependent variables included using students' achievement on these three tasks: conservation of number, length, and in identifying Euclidean shapes. Subjects individually were given a pretest over a period of one week before the study took place. The postest was administered in the same fashion as the pretest but at the end of the study.

The control group consisted of 40 kindergarten students who received just the regular daily lesson. The treatment group consisted of 40 kindergarten students who learned Logo programming for approximately 75 to 80 minutes per week. The results of the pretest were analyzed using the Chi-square. The analyses showed that there were no significant differences for the three Piagetian tasks. The results of the pretest on drawing Euclidean shapes by level revealed that most students were at the concrete operational level. Therefore, the test of drawing basic

Euclidean shapes was not used as a posttest. The posttest results on the conservation of numbers using Kolmogorov-Sminrov with a calculated Chi-square of .128 ($D_{(max)} = 1.50$), showed no significant difference. The posttest results on the conservation of length by levels for children revealed no significance. Chi-square was .11 ($D_{(max)} = 1.72$).

The researchers of this study found that learning Logo did not accelerate the cognitive development of five to seven-year-old children.

The sample was not randomly selected, and there was no information on the selection of the sample. Moreover, there were only 15 Apple computers available for 40 students. Thus, a group of three students had to share each computer, and this may have caused some students to be off task. The study did not mentioned whether one teacher taught both groups or there was one teacher for each group. ANCOVA rather than the Chi-square should have been performed because there were pretests and postests in the study. By using ANCOVA, the postest results could have been adjusted by the pretest. More, validity and reliability were not reported for the tests used in this study. Consequently, the children in this study were too young to expect a change in their cognitive development to take place.

Gallini (1987) studied the effects of Logo programming and CAI that enhanced a particular set of cognitive outcomes. The study concentrated on two types of cognitive outcomes: the ability to execute directions and the ability to formulate directions. Forty-four fourth grade students were selected from urban school districts in the southeast. The subjects varied on their achievement level based on their teachers' rating, and standardized achievement test scores (statewide basic skills test). There were two treatments in the study, Logo and the CAI. In each treatment there were 22 students randomly assigned. Each treatment group received instruction for 25 minutes three times a week at the end of the school day. The study lasted for five weeks.

Subjects in the CAI treatment group received instruction using educational computer software. The software concentrated on flowcharting skills and programming activities. The programming activities required students to identify tasks, breaking the task into smaller parts and writing a program to instruct the computer to execute each task. The educational software implemented in the study for the CAI treatment were Koala Pad and Rocky's Boots. The subjects in the Logo treatment group learned Logo programming. After each session students were given activities that required writing Logo programming to draw different geometrical figures. Subjects were given a pretest and posttest based on a ten-item scale. The test items concentrated on two tasks completed by participants in both the Logo and CAI treatment groups. The two tasks were following and formulating directions. Following directions was a technique that required subjects to follow or execute instruction step-by-step to reach the final product. Formulating directions required students to describe how a figure was constructed, completed, or modified.

The study indicated that the test had a content validity, established by a panel of five judges (fourth and fifth grade teachers in math and computer applications). The panel agreed that all of the items of both the pretest and the posttest matched the objectives of the study.

A pilot study was conducted on nine students drawn from the same population to measure the equivalence of test forms. Subjects were randomly given one of the two test forms, the KR-20 pretest and posttest reliabilities were .78 and .86 respectively.

A 2 by 2 mixed ANOVA model with repeated measures on the testing factor (pre/post) was performed. The analysis revealed that there was a significant main effect for the treatment factor, $F(1,42)=17.0, p<.01$. This result supported the hypothesis that different types of computer environments may be more effective in nourishing certain types of learning outcomes. More, the analysis also showed that the repeated factor was significant, $F(1,42)=14.68, p<.01$, and this result revealed an overall gain from the combined group from the pretest to the posttest. Interaction was non-significant.

A one-way MANOVA was conducted on the scores. The analysis found a significant overall multivariate treatment effect was observed, $F(4,39)=3.27, p<.05$.

Results of the study revealed that the Logo group achieved higher in ability to formulate directions than the CAI group.

The study did not provide information on the selection process of the sample. Furthermore, the researchers indicated that students were varied on their achievement levels based on their teachers and standardized achievement tests; however, there was no information such as validity and reliability about these standardized achievement tests.

The test item scale was viewed to have a content validity. However, content validity was not enough to support the validity of this instrument. Accordingly, construct validity was needed to support the validity of the instrument.

Hunter, Theresa, and Hyslop (1987) attempted to investigate whether a curriculum utilizing Logo would support greater cognitive growth than the current approaches. The study selected six elementary schools within Area 2, of the Calgary Board of Education. The selection of the school was based on the interest of the principals and staff along with availability of computer and Logo knowledge. Grades 3 and 5 were selected from these schools because the two units of these grades were felt to be thoroughly developed, and had sufficient maturity to undergo group and individual testings. The study consisted of three groups: Learning problem solving with Logo, learning Logo in traditional manner, and a control group. This study lasted fir six months

The learning problem solving with Logo group consisted of 76 students from three schools: Alex Munro had 24 students, North Haven had 26 students; and Vista Heights had 26 students. In this group teachers were given inservice training in Logo programming to facilitate the learning of problem solving using Feurestein's concept of mediated learning.

Learning Logo in traditional manner group consisted of 78 students who came from two schools: Rundle school which had 27 students, Colonel had 24 students from the first class and 27 students from the second class. In this traditional group teachers were given an outline of a specific Logo concept that needed to be covered during the treatment period.

The control group consisted of 23 students who came from a school called Cambrian. In this group the teacher use others types of problem solving that were normally carried out as a part of the curriculum (not computer programming).

All subjects involved in the study were given a pretest to measure their cognitive abilities. The Canadian Cognitive Abilities Test (CCAT: Version 3; Form A) was administered as the pretest. All of the verbal and non-verbal tests plus selected subtests from the quantitative sections were given at the grade 5 level. The results for students in grade 3 showed some improvements in the means of the pretest to the posttest, but no significance on the selected verbal subtests of the CCAT.

The results for Grade 5 showed that there were some improvements for all three groups on the verbal subtest of CCAT. The significant differences in magnitude of the improvements were not obtained. In the quantitative part, there were no substantial pre to post-test changes noted for either group.

The ANOVA was performed between the three groups in the non-verbal section of the test. The analysis revealed a significant difference different F(2,147)=4.108,p>.05. A t-test was conducted to determine the source of significance identified by the analysis of variance. The analyses of the t-test found significant. t's(2.167 and 2.388) were obtained for comparisons between learning problem solving with Logo, learning Logo in traditional manner, and the control groups. In such case the superior performance was by the learning Logo in traditional manner group. The mean improvement from pretest to posttest in the learning problem solving with Logo was 2.754, while in the learning Logo in traditional manner was 6.652 and in the control group was 1.15.

The researchers found that learning Logo in traditional manner group significantly achieved higher in non-verbal section than the thinking with Logo group and the control group.

The study indicated that students in grades 3 and 2 could study these two units that they were sufficient mature to undergo individual testing. Here the study did not provide evidence that these children reached this level of maturity and development or what they meant by evidence of maturity.

The control group included 23 subjects while other groups such as the learning problem solving with Logo group and the learning Logo in traditional manner group had 76 subjects and 78 subjects respectively. Therefore, the control group had a small sample. The larger the sample, the more likely for significant results.

The Canadian Cognitive Abilities Test was administered as a pretest and posttest to all subjects involved in this study. The reliability and validity of this instrument were not mentioned.

More, ANCOVA rather than ANOVA should be used for analyzing the data. Also the study used the number of students as the unit of analysis because in this study schools were selected therefore the unit of analysis should be the number of schools.

Problem Solving

In this section eight studies are reviewed and analyzed. These research studies examine the relationship between learning computer programming and problem solving. The measurement outcomes included problem solving skills, processes, and ability.

Swan and Black (1988) studied the relationships between learning Logo programming and the development of problem solving skills. Researchers in this study identified six particular problem solving strategies. These strategies are subgoal formation, forward chaining, backward chaining, systematic trial and error, alternative problem representation, and analogical reasoning.

Subgoals formation refers to breaking the problem into two or more simpler problems and then solving each subproblem in relation to larger problem. Forward chaining involves working from what is given in a problem towards the problem goal in step-by-step, transformational increments that bring one progressively closer to that goal. Backward-chaining focuses on the goal state and tries to deduce a preceding state from which that goal could be derived, then a state from which that state could be derived, and so on, working backward to what is given in a problem. Systematic trial and error involves the recursive testing of possible solutions in a systematic, guided fashion, and the problem reduction and the refinement resulting from such tests. Alternative representation refers to generating an alternative problem specification and then testing whether the problem specification suggests problem solution. Analogy involves these steps: (1) specify the desired goals by identifying the base and the target system; (2) perform a mapping between the base and the target systems; (3) test the soundness of the match in terms of both structural similarity and practicality; (4) repeat step 2 through 3 until adequate representation is discovered.

The study had three hypothesis: (1) when students applied these problem strategies in Logo environment, they would transfer these problem strategies to non-computer domains; (2) students would attain a developmental difference in their abilities to acquire and transfer problem solving skills; (3) there would be a difference in students' abilities to transfer problem solving skills depending on the base context(s) in which these skills were acquired.

The subjects of this study were 133 students in the 4th-8th grades of a private suburban elementary school. All students in this study had at least 30 hours previous experience in Logo.

All subjects were given a pretest and posttest on the ability to solve problems by implementing each of the six problem strategies. The researchers designed a separate measure for each of these problems strategies. Students were randomly assigned by grade to one of the three contextual groups to receive graphics, list, or both graphics and lists problems, respectively. A consistent instructional sequence was given on each strategy.

The students were introduced to each problem solving strategy through whole group activities designed to provide concrete off-computer models of the cognitive processes involved in them. Students worked on problems during two 45-minutes class periods per week for approximately 12 weeks.

A significant difference was found between pretest and posttest scores on measures of all strategies except backward chaining ($p<.001$). These results supported the first hypotheses of the study. The analyses also found significant differences between grade level on subgoal formation, systematic trial and error, and alternative developmental difference in students' abilities ($p<.01$). No significant difference between contextual groupings was found that supported the third hypothesis. The central finding of this study was that there was a positive relationship between learning Logo programming and most of the problem solving strategy.

The study did not provide enough information about the social-economics status of the subjects. In addition, the study did not indicate how the sample was selected. Moreover, these students were from a private school and had previous experience in Logo programming, and thus the results may not be generalized to the whole population of elementary students. More, the study did not provide enough information of what students learn in each treatment.

There were six problem strategy tests that were designed by the investigator; however, there was no reliability nor validity mentioned about these problem strategies tests. The results revealed significance (p<.001) between pretest and posttest scores obtained on the measures of all strategies, except backward chaining, across both contextual groupings and grade levels. However, the researchers did not indicate the name of the statistical method used to analyze the data. Without knowing the name of the statistical method it is unknown how these results were derived or if the results are legitimate. The analyses of the results also found significant (p<.01) differences between grade levels on measures of subgoal formations, systematic trial and error, and alternative developmental differences in students' abilities to acquire and transfer to particular problem solving strategies. Again the name of the statistical method was not mentioned.

Swan (1989) investigated the relationship between Logo and learning problem solving. This study addressed two questions:

1. Is practice in particular problem solving strategies superior to the discovery of learning supporting the acquisition

and transfer of problem solving strategies within Logo programming environments?

2. Is the Logo programming environment supportive of the acquisition and transfer of problem solving skills?

The study focused on implementing five problem solving strategies: subgoal formations, forward chaining, systematic trial and error, alternative representation, and analogy.

Subjects were 100 students from grades 4-6 grades of a private suburban elementary school. All subjects had at least one year prior experience in Logo. The subjects were given pretest and posttest on their ability to solve problems requiring the use of each of the five problem solving strategies.

Problem solving strategy tests contained sets of problems that required the application of the particular strategy being investigated. Two different versions of each test were used and randomly assigned by condition on the pretest. Students were then given alternative forms of each test on the posttests.

In the subgoal strategy test, students were measured on their ability in solving mathematical word problems. Students were asked not only to solve the problems but also to show how to break them into parts. Subjects were tested on the forward chaining skills by using a paper and pencil test version of the computer program called Rocky's Boots. In Rocky's Boots, symbolic AND, OR, NOT GATES were combined to produce machines that responded to targeted attributes and sets of attributes (e.g, blue diamonds, crosses or green circles, etc). A combination of gates could be built up in a forward chaining manner to achieve correct solutions.

In the systematic trial and error strategy test, subjects were tested on different symbol combinations to attain a coherent decoding system. The researcher chose two decoding exercises to test subjects' abilities to systematically utilize trial and error strategies. The first of these strategies was a shifted alphabet code, and the second involved variations on a number code problem. In the alternative representation strategy test, students tested the ability to create alternative representation using the Torrance Test of Creative Thinking. In this measure, students were given a set of circles and asked to use these as a basis for producing as many interesting and unusual drawings as they could. In the analogical reasoning strategy test, students were tested on the completion exercise which contains items representing both vertical and visual analogy. Students were then given one analogy and asked to complete a second according to the relationship involved in the former.

All subjects worked in pairs during their regular computer classes. These classes met for two 45-minute periods each week. The entire intervention took approximately two and one half months. Each subject was randomly assigned by grade to one of three treatment conditions, Logo graphic condition: a cut-paper manipulation condition, or a discovery learning; and Logo project condition. Students in the first two conditions received the same basic problem solving instruction but differed in the practice environment. Students in the Logo graphics group received practice problems involving Logo graphic programming, while students in the cut-paper manipulation group worked on a similar problem which used the cut-paper manipulation. Students in Logo projects learned Logo programming but not problem solving.

A one-way analysis of variance was performed on the means of the pretests. The analyses found the means to be statistically equivalent F(2,97)=0.33,p<.10.

ANOVA showed one between-subjects factor, treatment group; and two within-subjects factors, test and strategy. Significantly different factors were group, F(2,97)=12.81,p<.01; test, F(1,97)=5.94,p<.05; and strategy, F(3,97)=207.11,p<.01, indicating significant differences along all three dimensions. These results favored the Logo projects condition which had an overall mean score of 60.2%, compared with cut-paper manipulation and the Logo projects groups with means of 45.4% and 47.1%, respectively.

Tests by group interaction were analyzed by assessing the simple test effects at each level of the group. The analyses of ANOVA found there was a test effect for the Logo graphics group (F(1,97)=29.95,p<.01), indicating significant pretest to posttest changes among students receiving treatments, but not for the two other groups. The comparison between group means showed that students in Logo group graphics improved an average of 11.1% in points on the four measures, while the scores of the other groups either remained the same or declined. The analyses of the results of the simple test effect at each level of strategy where the subgoal formation measures were administered only once showed that the majority of students exhibited significant pretest to posttest differences (F(2,97)=11.59,p<.01). To examine the tests by group interaction, the simple test effect was assessed at each level of the group. A significant effect was found for the Logo graphics group, (F(1,97)=18.91,p<.01), whereas no significant effects were found for the cut-paper manipulation group.

The findings support claims that learning Logo programming helps students in problem solving where neither discovery learning nor direct instruction with concentrate manipulation practice can.

The study did not provide enough information about the subjects such as socio-economic status. Also, there was no information on the selection of the sample. The study did not report the validity and the reliability of the subgoal strategy tests. Moreover, the subjects were measured on the forward chaining skills by using a paper-and-pencil test version of the computer program called Rocky's Boots, but the reliability and the validity of this measure was not reported.

In the systematic trial and error strategy test, subjects were tested on different symbol combinations to obtain a coherent decoding system; however, again the reliability and the validity of this measure were not reported. Moreover, on the alternative representations strategy test, students were measured for the ability to create alternative representation, and again there was no reliability or validity reported. The same applied to the analogical strategy test; reliability and validity were also not reported.

The data of the study were analyzed using ANOVA; however, the study did not provide the mean and the standard deviation of the data for comparison. More, ANCOVA rather than ANOVA needed to be performed to adjust for the results of the pretest. To examine the tests by group interactions, the simple test effects were assessed at each level of the groups. There were no details about this simple test; it was unknown what this test was supposed to measure.

Macallister (1985) studied the relationships between programming and problem solving activities of young children. Data were collected on a larger group (19 students) using two mathematics tests. One test was given at the outset of the project, while the other was given at the end of the project. The two math tests were based on the written section of the math diagnostic arithmetic test and self-concept inventory (the scale of student's perception of ability). The ability scale had 70 items with five subscales for the child's self-perception of "general ability," "arithmetic," "school satisfaction," "reading and spelling," and "confidence." Data were also collected from the teacher's ratings on reading and spelling skills, a self-concept inventory, a record of time spent on the two computers in the classroom, and three paper and pencil programming tests.

Three programming measures that differed in the nature of their demands on the problem solving skills and programming knowledge were used. The first measure, "program reading," required students to predict the output of a given program. The second measure, "program writing," asked students to write programs for a given picture on the screen. The third measure, "design program," asked students to write programs of their choice for their own drawings. There were two sets of results that highlighted how programming was learned in this classroom, the amount of time students spent on the computer and the relationship of programming to the academic skills and self-concepts.

The second and third grade classes. They used a Commodore PET computer for games and computer assisted instructions. A Texas Instrument computer (TIC) was used for Logo programming. There was only one TIC available for all students to use during this study. This study lasted for six weeks. The children were taught basic Turtle graphics. Most of the instructions were given during a weekly morning session with the whole group, or the instructor worked with the children in small groups of three to five. The results showed that there was a significant correlation coefficient between the (non-Logo) PET and (Logo) TIC with the first math measure and teacher's rating of reading and spelling ($p<.02$). The analyses of the data found that there was a negative correlation between the TIC time and the programming measure ($p<.05$). There was no significant correlation between

PET time and TIC time with the first math measure or the teacher's rating of reading and spelling skills, and in fact it was negatively correlated (p<.05) with the second math measure. TIC and PET time were negatively correlated (p<.01), and TIC time was positively correlated with the programming measures (p<.05) yet time was not. The results showed that the time was negatively correlated with the program writing (p<.05). There was a significant correlation between TIC time and both the measures and teachers' rating of reading and spelling skills (p<.01).

The programming measures were all positively correlated with one another (p<.01). The total score for the programming measure was positively correlated (p<.05) with the total for self-perception, reading, and spelling. There was also a significant correlation (p<.01) between the programming total scores and both math measures, and the teacher's rating of reading and spelling skills. The researchers found that students who succeeded in programming generally succeeded in other areas.

The researcher did not provide information about the selection procedure of the sample. In addition, there was no information on the socio-economic status of the subjects.

The researcher failed to report the reliability for each instrument that was used in this study. Importantly, the study also failed to report the validity for each instrument used.

Children used a Commodore PET which was a computer for games and computer assisted instruction. It was unknown what students learned from these games and computer assisted instructions. Only one computer was used for teaching Logo; this one computer may have not provided students with enough time to learn Logo.

The correlation coefficient and the t-test were performed on the data, but the study did not provide the correlation (r) value, t-values, or alpha level. Also, it was not clear if the study used the number of students or the number of classes for the unit of analysis.

McGrath (1988) studied the relationship between the transfer of problem solving and the following: high school students learning a second programming language, those students learning a first language, or those receiving no programming instruction. Six classrooms were selected for this study; five of the classrooms were from two medium size towns in Illinois, and one classroom was from a small town in Wyoming. The study had five groups and lasted for one year (two semesters).

Groups 1 and 2 were two BASIC classes that included students learning their first programming language. In each class there were a few students who took Pascal as their first language on an independent study basis. Two BASIC classes were taught by the same instructor. Group 1 consisted of 21 students; 19 of these students learned BASIC, and two of them learned Pascal. Group 2 consisted of 20 students, 15 of these students learned BASIC, while five of them learned Pascal.

Two Pascal classes included students learning their second programming language; these classes were Groups 3 and 4. Group 3 consisted of 29 students; 20 of these students learned Pascal, while 9 learned BASIC as a second language. Some of the students in Groups 3 and 4 were also members of Groups 1 and 2. Group 5 consisted of 23 students who did not learn a computer language.

The dependent variables were the scores on these problems. Most of the problems were taken from standard problem solving sources; a few problems were designed by the researchers. The problem solving tasks contained six parts. Part one was the analogy, where students were given two problems separated by several other problems. Students was given the solution of first problem so they solve the second problem by making an analogy to the first problem. Part two was the re-useable procedure, where students were given a pair of problems that differed from the analogous problems above in that the procedure to be re-used in the second is identical to that used on the first but a new context. Part three used creativity, where two problems were given out of the traditional cognitive psychology literature that involved restrictive and innovative use of materials. In part four, students were given recursive problems like the Monster problem (a problem similar to the tower of Hanoi problem). This problem contained three monsters, each monster had five hands and was holding a globe. Students followed certain rules in shuffling the globe around until the large monster held the large globe, the medium monster then held the medium globe, and the small monster held the small globe. The fifth part of the problem task was the biconditional rule, where students were given shapes varying in size, color, and shape were shown categorized into two groups. A series of test shapes was given in which the subjects was to classify each belonging in either groups. Part six was debugging, where subjects were given a mystery presented with a set of eight suspects, seven clues, and the inferences the inspector made from each clue. Students were asked to figure out where the inspector went wrong and how she/he ended up with the wrong person as the murderer.

All the students were given a pretest during the third week of the first semester. The paper and pencil pretest consisted of 10 items. Nine items were given in the same six categories described in the problem-solving task. Out of curiosity, a tenth problem was included. An ANOVA showed that these groups had no significant differences at the outset (F=.94 p>.39).

During the second semester, Groups 2 and 4 were given a two-week problem solving intervention. The other groups held classes as usual. In the two intervention groups, the teachers talked explicitly about problem solving and things they could learn about solving problems from what had been learned in the programming. Creative problems were assigned every day, and the solutions and tactics were discussed. Three pieces of publishing software were implemented during these two weeks of intervention: Where in the World is Carmen Sandiego, the MECC problem solving program, and Rocky's Boots. All subjects were given a posttest at the end of the second semester. The results showed that there were no significant differences on any of the measures. Also, there were no significant differences between any of the measures in re-useable procedure problems.

The comparison between intervention and non-intervention students in the first language showed a significant difference t(39)=1.91,p<.05, in favor of students receiving the intervention. The means were 1.15 and .71, respectively. No other tests of creativity problems were found to be significant.

There were no significant results found in the recursive problem test. The analyses of the results for the problems using ANOVA and the number of languages learned revealed a significance. However, the planned comparison showed the intervention group performed twice as well as the non-intervention group in the two first language groups taught by the same teacher, t(39)=1.79,p<.05. The means were .50 and .24, respectively.

A two-way ANOVA performed for the debugging problem showed second language students to be significantly better at debugging than the first language students, F=5.59,p<.05. The mean for the first language students was .07, while the mean for the second language students was .23.

The researcher found that students receiving the problem-solving intervention in their first programming language course improved on the use of the biconditional and on creative problem solving. Furthermore, the researcher also found that for debugging students learning their second language significantly outperformed those learning a first language.

The study did not provide enough information on the selection process of the subjects. Information such as subject socio-economic status was desired. The unit of analysis used here was the number of students. Because there were five classes selected, the unit of analysis should be the total number of classes, which was five. The problem solving task implemented in this study was to measure subjects' abilities in problem solving. The researcher failed to report the validity and the reliability of this measure.

There was a significant difference (t(39)=1.91,p<.05) in favor of students receiving the intervention. The means were 1.15 and .71, respectively. This significant result was not of practical value because the difference (.44) between the two means was low and if the correct units of analysis were used the significant results will despair.

The study found significant results but did not indicate between which problems these results were significant. Post-hoc analysis needed to be applied on the significant results to determine where it was significant (between which problems there was significant difference result). Also ANCOVA instead of ANOVA need to be used for analyzing the data.

Blume and Schoen (1988) conducted a study to investigate the problem solving process used by eighth-grade programmers and non-programmers. This study hypothesized that the programmers used more techniques than non-programmers. These techniques were: systematic approaches, planning or preparation processes, used variables and equations more effectively, and more frequently detected and corrected more errors.

Six classes were selected, three of these classes were from fall term, and the other three were from spring term. All classes were selected from a junior high school in a Midwestern city. The subjects were eighth-grade students of a BASIC programming class. Fifty-eight subjects were in three classes that were offered in the fall, and 33 subjects were in three classes offered in the spring. Nine students from the spring classes were excluded because they had prior programming experience.

The Iowa Test of Basic Skills (ITBS) Problem-Solving test had been administered the previous Octcber. Quartiles for ITBS were constructed from the scores of all the students. These quartiles reflected the students' prior problem solving achievements. The quartiles were ranged from 1 to 4 (with level 1 highest).

There were three instruments implemented in this study: interview problems, written word problems, and written logic tests. Twelve problems were pilot tested in interviews with eighth-grade students in schools not involved in the study. These 12 problems were designed to measure students' abilities in problem solving techniques. Five of these problems were chosen for use in the interview because they elicited a wide range of processes in problem solving techniques.

The Iowa Problem-Solving Test was implemented to measure students' abilities to understand word problems and to apply specific problem-solving heuristic. This test had 15 multiple-choice items, and KR20 reliability was .72.

The researchers developed 10 multiple-choice items and eight open-ended items to measure students' abilities to apply the sequential logic that computer programming often entails. The KR20 reliability of the Problem-Solving Test was .81.

Students who enrolled in BASIC classes for the fall term were interviewed during the last week of the term, while students who enrolled in BASIC for the spring term were interviewed during the first week of the term just before the contents of BASIC programming had started. Each student was interviewed individually for 20 to 40 minutes. In each interviews, student was asked to "think aloud" while solving each problem. The interviews were audiotaped, the interviewer took notes, and each of the subject's written work was collected. Two coders independently coded each subject in solving the problems. Intercoder agreement ranged from .84 to .95 on the five problems. Using these codes along with the tape, transcript, interview notes, and student's written work, the two investigators jointly reached a consensus code.

A 2 X 4 X 5 MANOVA tested the significance of the group main effects and interactions of group X problem, group X level, and level X problem for the set of 17 dependent variables. The group X level X problem was not significant, F(204,1898)=1.12,p<.13, using the Hotelling-Lawley trace; thus the two-way interactions were directly interpretable. Significant multivarite F-values were found for the level X Problem interaction, F(204,1898=1.19,p<.0001; the level main effect, F(51,86)=2.10,p<.001; and the problem main effects, F(68,634)=18.76,P<.0001.

The multivariate F's were not significant for the group X level interaction, F(511,86)=1.11,p<.33; the group X problem interactions, F(68,634)=.93,p<.64; and the group main effect, F(17,30)=1.02,p<.47. These analyses showed that usage across all 17 processes did not vary significantly by a group or by the interaction of groups with either level or problems of both.

Three univariate main effects (systematic trial, checking and correcting errors) and one interaction involving group (group X level, for impasses) were significant at the .05 level. The F-values were 4.98, 6.28, and 5.16, respectively.

The researchers found that programmers used systematic trial more frequently than nonprogrammers did. The researchers also found that programmers also checked for and corrected more errors in their potential solutions.

The study did not provide enough information about the subjects, such as socio-economic status, and also there was no information provided on how the sample was selected. The study used the number of subjects as the unit of analysis. In this study classes were selected, and the correct unit of analysis was the number of classes in each group, which was three.

The study failed to report the validity of the three instruments (interview problems, word problems, and logic tests).

The reliability of the word problem was not mentioned. Furthermore, the study indicated that the interview problems were pilot tested in the interview with eighth-grade students. However, the study did not provide information about the subjects used in the pilot test. Therefore, it was not clear whether both subject sets (the subjects in the study and the subjects in the pilot study) were from socio-economically similar schools.

The study found a significant difference in reporting the F-value and calculating the p value. It is preferable to set alpha level before conducting MANOVA and then calculate the p-value and compare it to the initial alpha levels. If the calculated p value was smaller than the initial value, then the result would be significant, assuming that the F-value is high enough to be significant.

The study conducted 17 tests in MANOVA. Because of the large number of F tests, there was a chance to make a type I error in the study.

Dalton (1986) compared the effects of Logo use with teacher-directed problem solving instruction and conventional mathematic instructions on the problem solving ability, basic skills achievement, and attitudes of junior high level students.

The subjects were 97 students selected from five classes of a seventh grade mathematics course. Three instructional treatments were implemented: a problem-solving strategy instructional treatment, a structured Logo treatment, and a control.

The problem solving treatments consisted of approximately 20 hours of instruction in problem solving strategies. Six problem solving strategies were used: Guess and Check, Make a Table, Patterns, Make a Model, Elimination, and Simplify.

The Logo treatment used approximately 20 hours of computer time in which students could explore the turtle graphics capabilities of the Terrapin Logo language. Each learner was provided with a lesson which contained a list of new commands and exercises and asked to independently complete the lesson. The control group was given additional time for completing any school assignments and/or recreational reading. The dependent variables were students' achievements, attitudes, and higher-level thinking skills.

There were two measures of achievement implemented in this study. The first measure was the Program Criterion Reference Test (PCRT); this test was used to measure student's mastery of grade level objectives. The test contained 80 multiple-choice items, and the spilt-half reliability coefficient for this test was 0.78.

The second achievement measure was the mathematics subtests of the Comprehensive Test of Basic Skills (CTBS). Using the data obtained from this study, the split-half reliability coefficient for the combined scales was 0.90.

Students' attitudes were evaluated with two measures. The first measure was the Revised Math Attitude Scale, a Likert-type scale questionnaire; the spilt-half reliability coefficient of this measure was 0.93. The second attitude scale was the School Attitude Measure (SAM), a Likert-type scale that contained 85 questions pertaining to attitudes toward school, teachers, and education in general. Data collected from this test indicated that the spilt-half reliability coefficient was 0.78.

There were two measures of problem solving skills used in this study, with the first measure being a Test of Cognitive Skills (TCB). The TCB consists of four sections: Memory, Analogies, Sequences, and Verbal Reasoning. Data from these tests yielded a split-half reliability coefficient of 0.88 for the TCB.

The Test for the Non-Routine Problem Solving Skills consisted of 20 items that measured the non-routine problem solving skills. These problems are open-ended in nature with several possible solutions, and the spilt-half reliability coefficient for this test was 0.76.

Students that were in the fifth and seventh grade mathematic classes were assigned to the three treatments. Students were then classified as high, average, or low in prior achievement based on sixth grade CTBS scores.

Each of the learners was subjected to the respective treatments for two instructional periods of approximately 45 minutes per week over a period of two months (20 sessions in all). At the end of the experimental periods, each learner was posttested on the dependent measures.

ANOVA revealed that there were significant differences in both the achievement level means and the treatment group means (F=7.22,p<.001 and F=32.08,p<.001, respectively). The problem solving groups differed significantly from the control group but not from the Logo group. The high group differed significantly from the low group but not the average group.

The results of the Test of Non-Routine Problem-Solving Skills showed that the means of the problem-solving groups were significantly higher (p<.001) than the means of the control and the Logo groups. However, the means of the Logo and control group did not differ significantly.

ANOVA was performed on the Comprehension Test of Basic Skills. The analysis showed that there was no significant treatment toward the main effect; however, the achievement means were significantly different (F=22.43,p<.001), specifically for the high group. But, the means of the average and low groups were not significantly different.

Using ANOVA, the means for the achievement level groups were significantly different for the PCRT (F=6.73,p<.001). However, there were no significant differences among the treatment group means.

The ANOVA was performed on the Revised Math Attitude Scale. This analysis showed that the means for the treatment groups were significantly different (F=22.32,p<.001), from both the Logo group and problem solving groups. The means were significantly greater than the control group. However, the means of the problem solving and Logo groups themselves were not statistically different (p<.05). In addition, the means for the low group differed significantly from the means of the high group (p<.05). However, the means of the low and the average groups did not differ significantly, nor did the means of the average and high groups. ANOVA analysis on the School Attitude Measures showed that neither treatment nor achievement group means differed significantly.

The results of this study suggest that the problem-solving skills fostered through Logo use may not transfer outside the context of Logo, since Logo apparently provides only a single algorithm which may not apply to many types of non-routine problems.

The study did not provide enough information about the subjects, such as the subject's socio-economic status or the selection process of the sample. More, the study did not mention the number of students in each treatment group. There was no information provided about the teachers who participated in the study. Actually, it was not clear whether one or more teachers taught the three groups. Also, there was a possibility of teacher effect because all teachers were from the same school, and there was a possibility that the teacher in one groups may have given students more attention than the teacher in the other group. Thus, the teachers need to be rotated to all groups to avoid these effects or they need to be observed.

The researcher failed to report the validity for each instrument used in this study. The study used a post-hoc method to determine where there was significant difference between groups; however, the study did not mention the name of the post hoc method used.

The study also mentioned that there was a significant difference between the means of the low group and the high group in Revised Math Attitude; however, there was no statistical value reported to support this significant result. Moreover, it was not clear from this study whether the number of students or the number of classes was used for the unit of analysis. The correct unit of analysis should have been the number of classes.

Rieber and Lioyed (1986) investigated whether young children given the experience of Logo programming would acquire skills in problem solving and basic geometry. Twenty-five students (average age=8.08 years) were selected to represent the experimental group. These students were from a regular public school classroom. Twenty-two students (average=7.82 years) were selected to represent the control group. All subjects were second grade students in intact classes but from different districts. Both groups were selected based on the willingness of their teachers to participate.

The experimental group was given one hour in Logo programming each week for three months. There were four computers available for the students and the rest of the school's population. Subjects were required to do a series of activities. In each activity a card contained a simple geometry shape such as a square. Students were asked to create a Logo program that drew the shape. The control group received the regular class content. Both groups received similar instruction in terms of content and used identical textbooks.

There were two dependent variables. The first one was the measure of the ability of students in problem solving; this measure consisted of two parts. The second measure was a test on geometry knowledge. The problem solving was based on two classical Piagetian activities used originally as examples of problem solving in individuals at the stage of formal operations. One activity involved a combinatorial task and the other was a permutation task. The measure of the geometric concepts contained the following parts: angle recognition, concept of angle, line segment, and the rotation of given figures.

The analysis of the pretest data of problem solving measure used t-test and revealed that there was no significant difference between the experimental and the control groups in the pretest. The analysis of the data for the experimental and control groups for the problem solving on the postest showed a significant difference, $t(45)=4.87, p<.01$; the means of the experimental and control groups were 17.6 and 10.59 respectively.

A t-test was performed on the pretest and the posttest of the problem solving measure for the experimental group. The analysis of the data showed a significant difference, $t(24)=4.9, p<.01$, the mean for the pretests and posttest were 11.52 and 17.6 respectively.

An additional analysis was conducted to compare the pretest and the posttest in the problem solving of the experimental group; this analysis showed a significant difference, t(24)=4.9,p<.01, and pretest mean=11.52, posttest mean=17.6. The analysis of the control group's pretest in problem solving showed no significant difference.

There was no significant difference between the experimental and the control group in the geometric mathematical ability. The experimental and control group's means for geometry test were 14.8 and 12.5 respectively. A comparison of these means using an independent t-test revealed a significant difference t(45)=1.78,p<.1.

The additional analysis of the experimental group's pretest and posttest of geometry abilities used the dependent t-test and showed a significant difference, t(24)=3.21,p<.01 where the control group showed no difference. The means for the experimental group in the geometry test for pretest and posttest were 11.4 and 14.8 respectively. The means for the control group in the geometry test for the pretest and posttest were 11.73 and 12.5 respectively.

According to the researchers the findings of this study tended to support the developer of the Logo language, Seymour Papert. Papert claimed that successful programming interactions encourages the development and exercise of problem solving.

There was not enough information about the selection process for this sample. The only information given was that the subjects were selected upon the willingness of their teachers to participate. Also, the study did not provide information about the socio-economical status of the subjects.

There were only four microcomputers which were shared among 25 students; thus the access time for each student to utilize the microcomputers was very short. More microcomputers must be used to give more access time for each student. Moreover, researchers need evidence that students are getting equal access to used the computers. Also, it was not clear whether the teacher in the experimental group had proper training in teaching Logo. The study also mentioned that the control group did not receive Logo treatment nor any access to microcomputers. Students may have had access to a microcomputer at their home or elsewhere.

The researchers failed to report the validity and the reliability of problem solving ability test and geometry test. A comparison between the experimental and the control group used t-tests to show significant differences. The result was not of practical value because the difference between the experimental and the control means was low. Moreover, the ANCOVA should have been used for analyzing the data instead of the t-test results.

Analysis of the experimental group's pretest and posttest of geometry abilities used the dependent t-test and showed a significant difference but this significant comparison is just between the pretest and posttest of the experimental group not comparison between the experimental and control group.

Horner and Maddux (1985) investigated the effects of Logo programming on problem-solving abilities: locus of control, attitude toward math, and the ability to recognize the size of geometric angles. The students who participated in this study identified themselves as learning disabled (LD) and non-learning disabled (NLD) at a junior high school in an urban west Texas school district.

There were four groups: two experimental groups and two control groups. Both the experimental group and the control group included one intact group of mixed seventh and eighth-grade LD math students and one intact group of regular eighth-grade math students. The number of subjects were 74; the experimental LD group had 16 students, while the experimental non-LD group had 21. The control LD group had 20 students, while 17 students were in the control non-LD group.

The data of pretests and posttests were collected on all subjects using four instruments: Group Assessment of Logical Thinking (GALT), Intellectual Achievement Responsibility Questionnaire (IARQ), Fennema-Sherman Math Attitude Scale, and the Horner Angle Recognition Test (HART). The HART test was a 28-item multiple-choice test that was devised by the researchers to determine students' abilities to recognize the size of geometric angles.

Subjects were given a pretest where the item on each instrument was read orally to the subjects to avoid the effects of poor reading. Students in the experimental group received Logo instruction for 14 sessions during their math classes, and each session lasted for 55 minutes. The control group received the regular math curriculum. To determine the effects of Logo on problem solving abilities, students were asked to complete a weekly attribution checksheet. Students were then asked to report if they felt successful or unsuccessful on the weeks' Logo activities. They were also asked to choose a reason for their perceived performance, the reasons that could be selected indicated attributions to effort, ability, task difficulty, or lack of chance.

Students were categorized either internal or external, based on their Logo activities. Those students who reported at least four activities out of six efforts and ability statements were classified internal. Those who had at least four out of six task difficulties and lack attributions were classified as external. The purpose of this classification was to determine if students' attribution classifications (internal or external) would be similar to their IARQ classifications.

A two-way analysis of covariance was performed on the posttest mean scores with pretest scores as the covariate. The analysis revealed no significant differences for problem solving, locus of control, math attitudes, or angle recognitions. The study found a significant difference between observed and expected attributions toward Logo. On comparing the expected and observed frequencies, there were 14 internal and 14 external IARQ categories and 26 internal and two external Logo attribution categories (N=28). A chi-square analysis was performed to test for significance. The result of the chi-square showed there were significantly more students categorized as internal, based on Logo attributions, than were expected (X^2=20.61, p<.01).

The study indicated that Logo instruction did not produce a significant difference for problem solving, locus of control, math attitudes, or angle recognitions.

The study did not mention how the sample was selected and the rational for selecting this subject (learning disabled and non-learning disabled). There were four instruments implemented in this study. However, the researchers failed to report the reliability and the validity for each instrument. In addition, there were no descriptions of what each instrument was supposed to measure. Also, the study did not indicate whether the Logo instructors had previous experience in teaching Logo or what did students learn in Logo.

There was a possibility of teacher effect, for they were from the same school. To avoid this effect, teachers needed to rotate while teaching both groups (experimental and control).

Specific Cognitive Skills

This section contains six studies that are reviewed and analyzed. These studies measure the effect of learning computer programming on specific cognitive skills. The following are the measurement outcomes of these specific cognitive skills: rule-learning task, development of thinking skills, reasoning skills, general variable skills, and planning skills.

Degelman, Free, Scarlato, Blackburn, and Golden (1986) examined the effects of a short-term single-keystroke Logo experience in rule-learning tasks on kindergarten children. The study consisted of the entire class of 15 kindergarten students from a private day-care center.

There were two groups in the study, seven children (six girls, one boy) were randomly assigned to receive the Logo experience, and seven students (three girls, four boys) were assigned to a wait-list control. The remaining child (a girl), received the Logo experience. The average age was 5.3 years. Two Apple IIe microcomputers with color monitors and a single-keystroke Logo program was implemented. A single-keystroke Logo program required student to type a single keystroke rather than a full command. For example, the single-keystroke "F 10" command caused the Turtle to move 10 steps forward while in regular Logo the same command was "FORWARD 10 or FD 10".

The Logo group students received 15 minutes of instruction per school day, for five weeks, with a trained experimenter present at all times to guide, observe, and provide help as needed. The children were free to create any design they wished in Logo. All subjects were given between two and four rule-learning problems to measure their logical thinking. After the testing of the children, students in the control group started a similar period of Logo training.

To measure the dependent variable, subjects were shown stimulus cards that contained elements varying on three dimensions: shape (square, circle), color (black, white), and size (large, small). A random sequence of the eight possible stimulus configurations was used for all four concept-learning tasks (large black square, small white circle, small white square, large black circle, small black square, large white square, small black circle, large white circle). This sequence was repeated six times to create a total of forty-eight stimuli for each of the concept-learning tasks.

Subjects were considered to successfully solving a given problem if they correctly responded to sixteen successive stimuli, or if they verbally identified the rule.

All students were given a test on problem 1 and 2. The results of these test showed that the Logo group had a mean of .87 while the control group had a mean of .71. The results of the t-test on the matching groups revealed significant differences (t(6)=2.96,p<.05). Therefore, students that received five weeks of Logo instruction had a significantly higher proportion on correct responses on two problem solving tasks. Another analysis was performed on the number of subjects in each group successfully solving each problem. Eighty percent of the Logo students solved problem 1 and 2, while 43 percent of the control subjects solved the first two problems. Of the Logo subjects who solved problems 1 and 2, 33 percent also solved problems 3 and 4. Of the control students who solved problems 1 and 2, 33 percent also solved problems 3 and 4.

The researchers concluded that children receiving Logo instruction achieved higher in problem solving task than children did not learn Logo.

This sample was small; subjects came from a private school, and might not represent the overall population. There were two microcomputers shared by 15 students. It was not clear whether these two computers were available for students for 15 minutes or more. If these two Apple computers were available for fifteen minutes, each student had a few minutes to have utilized the computer, and this was a very short time for students to practice Logo.

It was not clear whether the subjects in the Logo group followed a specific curriculum or objective. Students were free to create any design they wished to make. Also, the study did not indicate what the control group learned during the 15 minute session.

A random sequence of eight possible stimulus configurations used all tasks to measure the four concepts. However, the researchers did not mention the reliability and the validity of this measure. The researchers used the number of subject as the unit of analysis. Because all students came from one class, the correct unit of analysis was the number of classes.

Kurland, Pea, Clement and Mawby (1986) studied the development of thinking skills and programming ability of high school students. Their study was based on these three questions: (1) Do students who learned programming for two years achieve better in reasoning and thinking skills than students who learned one year of programming?; (2) Is there a relationship between programming, math and reasoning skills; in other words, are certain math and reasoning skills good predictors of the success of the programming course?; (3) Are students able to write advanced programs after their second year of programming?

The subjects of the study came from urban public high school with mixed ethnic and socio-economic status. Fifteen students who varied in ability according to their GPA's and grade level represented the experimental group. These students were 10-12th graders and were learning second year programming. Fifteen students were selected from a pool of volunteers who matched the experimental students in their math backgrounds and GPA's represented the control group. The control group was divided into two parts. The first part consisted of nine students who had no prior programming. The second part had six students who had some (one year) programming.

At the beginning of the year, students were given a pretest to predict their performances in programming classes. The pretests covered the following concepts: procedural reasoning, planning, and mathematics.

A set of tasks designed to measure procedural and conditional reasoning is called a Procedural Reasoning Test. This measure contained two tasks: non-verbal and verbal. The non-verbal reasoning task had two parts. In the first part, students were asked to determine all the goals that legally could be reached using a set of tokens. The second part had additional components that focused on students' planning skills. The verbal reasoning task contained two tasks. The first task was analogous to the non-verbal procedural reasoning tasks, but given in verbal form, while the second task had a complex conditional structure with a number of goals and conditions for satisfaction.

In the planning task students were asked to schedule a set of classroom chores. One of these chore orders was to clean-up a classroom in a short period of time. The chores were to be executed by a "robot." The robot responded to a set of commands and specified the amount of time it took to perform the specific actions.

The math test required students to understand the relations from a prose description of a situation and change this relation into mathematical terms.

Non-Verbal Procedural Reasoning Task Two contained two parts. In part A, students were shown a set of passes and were asked to find the goal they could reach with the passes. In part B, students were presented with a set of passes and were required to find the correct path leading to a particular goal.

The Debugging Task asked students to find the error "bug" in a set of driving instructions written for another person to follow.

The math test focused on calculating the values of variables and translating prose description into symbolic expressions.

Algorithm Design and Analysis Task contained two parts: analysis and the design. The analysis part asked students to make a plan, whereas the design part asked students to write an algorithm for the plan.

Programming Skill Measures measured the students' programming skills in the experimental group. The Logo Test was the second programming measure which was designed to measure programming comprehension and programming production.

The posttest wer administered at the end of the year. These tests assessed the cognitive skills gained from the programming course. The contents of this measure contained the following concepts: procedural reasoning, decentering planning, math ability, algorithm, and comprehension.

The ANOVA was conducted on all pretests by comparing the composite scores for each pretest measure. The results of the ANOVA showed no significant difference between groups of any measure.

The means and standard deviations for the math pretest scores showed that students had difficulty computing the value of variables except in the simplest cases. Students were unable to create a symbolic expression for a word problem.

The scores for part A and part B of the non-verbal reasoning tasks were low for each student. Students had difficulties in discovering some of the correct goals that were in part A.

For all of the verbal tasks, the performance indicated that all groups of students had difficulty following the complex nested conditionals given in the verbal instructions.

The results of the performances by each group on the planning task for the pretest showed that there were no group differences due to feedback conditions. The non-verbal procedural reasoning posttest composite scores were developed for part A and part B; there were no significant between-group differences.

The results of the debugging posttest showed that groups did not differ in their abilities to detect error "bugs" of any of the classes of bugs.

The ANOVA analysis of planning posttest revealed that there was no significant difference results between groups on any of the measures of plan execution time or planning behavior.

The ANOVA analysis of the math test showed no significant difference between groups on either the variables or the symbolic expression problems. A second analysis of variance was conducted on the math performance comparing the score on the subtest of those problems which were identical to the problems on the pretest. The repeated measure of ANOVA (group by session) showed that the posttest performance was significantly better $F(1,38)=2.25;p<.01$.

The results of the two parts of the algorithm design and analysis test were analyzed. The analysis showed no significant difference between the two groups.

The performance on the pretest and posttests with a composite of the test scores for each language and with subscores on the Logo tests were correlated. The procedural reasoning pretest scores and the math variable pretest correlated significantly with the programming score r=.66,(p<.01);r=.77,(p<.01). Of the posttests, the procedural reasoning, debugging algorithm, and math scores correlated significantly with the programming tests, and these correlations were r=.65,(p<.01), r=.63,(p<.01), r=.85,(p<.01), r=.77,(p<.01) respectively. The math variable pretest and posttest scores and the algorithm task scores correlated with the programming measures. The correlation of the comprehension and production parts of the Logo test and the pretest and posttest correlation were significant (r=.68,p<.01,and r=.69,p<.01, respectively).

The results of the programming skills test indicated that most students had difficulties in understanding the programming languages. Students did not demonstrate skills for such systematic reasoning in their programming, nor in their solving of the math problems.

The results of this study showed that even after two years of studying computer programming, many students had only rudimentary skills of programming. The study also found that programming skills do not appear to transfer to other domains which share analogous formal properties.

There was no information on the selection of the sample. The researchers of this study did not mention the validity and the reliability of each instrument that was used in this research. Without validity and reliability it is difficult to trust these instruments, and the results of the study.

The study used the number of students as a unit analysis, the correct unit of analysis was the number of classes in the study. Also the study found several significant correlations between reasoning pretest scores and math variables pretest correlated with programming score, and the postest scores of procedural reasoning, debugging algorithm and math scores correlated with programming tests, but these significant results were not of practical value because the r^2 was low. Moreover the ANCOVA rather than the ANOVA should been used to analyze the results. The study also mentioned some results were significant, but there was no statistical value or unit of analysis to support these results.

Clement, Kurland, Mawby, and Pea (1986) investigated the relationship between analogical reasoning skill and aspects of programming that involved mapping program structures across problems. Subjects were 17 years old (9-11th grade females) and enrolled in the Logo programming course. The Logo programming course was part of a six-week program designed to improve math skills. Students spent 90 minutes each day to learn Logo programming for a total of 45 hours.

The analogical reasoning task was adapted to measure students' abilities in determining the structural similarities between two story problems. In order to find the similarities between two problems they had to map the structures of one problem and its solution onto a second problem. The researchers developed two strategies in solving the problem. The first strategy was students' abilities to recognize spontaneously a base problem as potential analog and to target it, and the second one was students' abilities to carry through the structure mapping once analog was recognized.

The target story was Dunker's radiation problem. In this problem, the doctor needs to destroy a patient's tumor; however, the doctor can not operate. The doctor's choice was to use rays with sufficient force to destroy the tumor without harming the surrounding tissue.

The base story was about a Military general who wanted to capture a fortress surrounded by roads radiating outward. The general must use large troops of men to capture the fortress; however, the road was mined so that large troops would detonate the mines while small troops could travel safely. In order to capture the fortress the general sent small troops of soldiers down different roads so that they arrived at the same time at the fortress.

The analogous solution for this problem was to have the doctor to direct several weak rays on the tumor. These rays were directed from several different directions simultaneously so they converged in full strength on the tumor.

The task had two phases. In the first phase, students received the military problem and were asked to solve it. Then, they were asked to solve the radiation problem. The students had to read the stories and gave their responses in writing. For the second phase of the task, students were given the convergence solution to the military problem and were asked to solve the radiation problem in a similar way. During this phase, students were also asked to articulate the similarity of the two problems and the similarity of the solutions.

A program comprehension task was implemented to assess understanding of commands and flow of control in two graphic programs containing subprocedures linked by control structures. This program task was explained in another study. In this program task students were required to write for each program a short description of the purpose of each subprocedure and draw output of each program.

A program production task measured the strategic technique in writing programming that produced different geometric figures. In this task, students were shown a set of different geometric figures, five of which were structurally similar. Then, they had to choose five out of seven sets and write a program that would display these figures on the screen.

The analogical reasoning task was administered during the first week of the course. This task consisted of two phases. In the first phase, students were asked to solve the military problem. Then, students were asked to solve the radiation problem. In the second phase, students were asked to read a description of the convergence solutions to the military problem and write a solution for the Radiation problem. In second phase, students were also required to describe the similarities of the problems and the solutions to the two stories.

The result of the analogy task phase showed that nine of the 17 students gave the convergence solution to the base (military) problem. Six students gave an incomplete version of this solution; they divided the general's forces into small troops, but they sent them down a single road in sequence. Two students did not solve the problem.

The result of the second phase showed that only one subject spontaneously solved the target (radiation) problem using the convergence solution. Three subjects did not complete the solution, while the remaining subjects had incorrect non-analogous solutions. In the second part of phase 2, students were asked to describe the similarities of the problems and the solutions to the two stories. Eight of the students gave the complete structure mapping, while nine students gave a partial vague mapping. For the description of similarities of problem: five students gave complete structure mapping, eight students gave partial vague mapping, three students gave irrelevant mapping, and other subjects gave no mapping.

Students were given scores for programming production and comprehension tasks. For the comprehension production program students were scored on their written description of the function of each program and its output. In the production task, students were scored on the strategic skill in writing and running of the program. The results showed that students had relatively poor understanding of the purpose of the program. Students also had difficulty in writing programs.

The correlation analysis was performed to examine the relationship between the analogical reasoning task and the Logo tests. The only significant correlation was the reuse of subprocedures across programs ($r=.55, p<.01$).

The study provided evidence that the ability for analogical mapping is related to practice of identifying and writing subprocedures that can be used in various programming tasks.

The study did not provide enough information about the subjects of the sample. Information such as the socio-economic status and selection process of the sample were needed.

The study failed to report the validity and the reliability of the analogical reasoning test measure. Furthermore, a program production task was implemented as a strategic programming technique to write graphic programs. Again, the researcher failed to report the validity and the reliability of this measure. Also, the researchers referred to another study for more details on these instruments.

The analysis of the results showed that there was a significant correlation between the analogical reasoning task and the Logo tests. The only significant correlation was the reuse of the subprocedures across the program ($r=.55, p<.01$). This correlation is not of practical value because r^2 was low.

MaCoy (1988) studied the relationships between computer programming experience, mathematics experience, and general variable skills. Forty-six students (aged 9 to 17) at a summer computer camp participated in this study. Their ability levels were varied, as their backgrounds in both computer programming and mathematics.

A placement test on programming was performed to measure students' knowledge in computer programming. This test consisted of three programming problems to be completed in either BASIC or Pascal.

The mathematics experience was defined as the number of years of higher mathematics courses had successfully completed. This information was collected from the participants.

The general variable skill test was also implemented. This test had 15 multiple-choice items, where verbal situations were described and the students selected the equivalent verbal expressions.

The results revealed that both computer experience and mathematics experience were significantly correlated with the general variable skill; correlation coefficient values were .463 and .555, respectively. Further results showed that when the general variable skill was regressed on the two experienced variables, computer programming experience was a significant predicator and the mathematic experience was not, R^2 =.35.

The researcher reported that both computer experience and mathematics experience were significantly correlated with general variable skill. The researcher also found that the relationship of computer programming experience with general variable skill was stronger than the relationship of mathematics experience with the variable skill.

The study did not provide information about the socio-economic status of the subjects, nor did it mention how long the study lasted and also there was no information provided on the selection of the sample. Because the study was conducted on camp placement, thus findings of this study can not be generalized to typical school setting. The camp placement test was implemented to measure the students' experience in computer programming. However, the study did not report the reliability and the validity of this instrument. Also, the researcher failed to report the reliability and the validity of the General Skills Test.

These results revealed that there was significant correlation between computer experience and mathematical experience. However, the study did not indicate the alpha level, or the unit of analysis, and the r^2 was low and it was not of practical value is questionable. Moreover, the study did not provide the alpha level for the regression analysis and the unit of analysis. In addition to that, the R^2 value of the regression analysis was too low, and it is not of the practical value.

Pea and Kurland (1984) analyzed the relationships between learning Logo programming and planning skills. Thirty-two students from a private school in Manhattan were selected for this study. The first half of the subjects were 8-9 years old while the other half were 11-12 years old. Subjects were selected based on their teacher's willingness to participate. The experimental group consisted of four boys and four girls (8-9 years) and four boys and four girls (11-12 years). The control group had four boys and four girls of each age. The experimental group received Logo instruction for two 45 minute sessions per week, while the control group did not receive any treatments.

Students in the experimental group were selected based on two factors: (1) the amount of time they spent working with Logo prior to the study and (2) teacher assessment of reflective and talkativeness of subjects. The students in the control group were selected based on the second factor.

A digit-span task and WISC Block Design subtest were administered to measure students' cognitive styles. Cognitive style is a different way of perceiving and organizing information. The results of these measures showed that the experimental group and the control group did not differ in their scores.

The researcher designed 22 inch by 30 inch plexiglass map with a scale of 1 inch to 15 inch. This map contains a fictitious classroom as a task. Another map, a replication map was used for the second task. This map differs from the original map because its orientation transferred 180 degrees. There were five major chores that students were required to accomplish in a minimum of 39 distinct chore acts. These five chores were (1) watering two plants, (2) erasing and washing two blackboards, (3) feeding a hamster, (4) putting away objects (returning and washing paintbrushes).

The treatment consisted of two sessions where the planning task was administered at the beginning of the first session and at the end of the second session. Between-participant group variables were: (1) Group (Logo, non-Logo), (2) Sex (male, female), and (3) Age (younger, older). The key within-participant variables were; (1) scores for first-last plans within sessions and (2) scores for session one versus session two.

Each child was tested individually by taking the child to a filming room to be seated at table with the plexiglass map upright on an attached stand. The child was asked to make up a plan that would accomplish many of the classroom chores in a short spatial path. The child was asked to think out loud while planning. The same procedure was followed for the second session.

The videotapes were transcribed and the number of plans for each child was recorded. The results showed that the mean number of plans per child was 3.64, and there were no significant group or age differences in this respect. The study found that the route efficiency scores were significantly increased within age from the first session to the last session within session. The mean session route efficiency score across age had increased from 65 to 80 out of 100 points. The study also found that the Logo group did not differ from the control group for the route efficiency scores, at either age, for plan or session. The study also indicated that no significant correlations were found in WISC, digit-span, and the route efficiency scores.

Children's plans were analyzed in terms of plan features. The study found that there was a significant correlation between the first and the last sessions; the correlation coefficient ranged from .66 to .72.

The mean score for the plan significantly improved for each group across plans and session. However, the Logo group did not differ from the control group on any of these comparisons.

The researchers tried to determine the degree of flexibility of a child's decision during the planning process in two ways: (1) looking at the number of transitions between decision making during each plan, and (2) looking at the number of transitions made between levels of decision. The study found that the number of type transitions for each plan correlated with the mean number of level transitions in each plan. The experimental group did not differ from the control group on these comparisons.

The researchers found that students who had spent a year programming did not differ on various developmental comparisons of the effectiveness of their plans and their processes of planning than same-age students who had not learned computer programing.

The results of WISC (cognitive style measure) and digit span (processing capacity) showed that there were significant correlations between these measures. However, the researchers did not report the correlation coefficient values, unit of analysis or the alpha level.

The researchers failed to report the validity and the reliability of the digit-span task and the WISC Block Design subtest. Videotaped sessions were transcribed; however, the interrator agreement was not mentioned. Moreover, the study did not mention the name of the statistical method that was used to analyze the results that were derived from the videotapes. Thus, it was un clear if the appropriate statistical method was used.

The analyses of the results from the map indicated that the route efficiency score significantly increased with age from the first plan within the session. However, the study did not provide the name or the result of the statistical method that was used to show these results or if they were significant.

The results found that the mean cluster and plan efficiency scores were correlated for the first and the last (r's ranged from .66 to .72). Consequently, this correlation was not of practical value.

The study indicated that the mean for the plan score was significantly improved for each group across the plans and the sessions. Again, the study did not provide statistical values to support this claim. As for the difference in the type of planning decisions made for the first versus last plan, the results revealed that only in session one did the children make significantly higher-level decisions in their first plan than their last. The older children produced more high-level decisions than did younger children. However, the study did not provide evidence such as statistical methods or unit of analysis, to support these results.

The second study was similar to the previous one and conducted by the same researchers, Pea and Kurland (1984). Their aim was to study the potential programming effects on planning skills more closely. This study took place one year after the first study, in the same school, and with the same teachers.

Half of the subjects were 8-10 years old from 3-4th grades, while the other half were 10-11th years from 5-6th graded. The teachers in this experiment took more directive roles in guiding their students in learning Logo than in the previous study. Teachers gave weekly group lessons and demonstrated key computational concept and technique.

The study followed the same general pre-post design of previous study. Thirty-two students participated in both sessions. In the second session an additional 32 students were tested in order to produce a more sensitive test for planning task analysis. The control group contained 32 subjects (two combined 3-4th grades and two combined 5-6th grades). The experimental group consisted of four boys and four girls of each age while the control group contained four boys and four girls of each age. The experimental group learned Logo programming while the experimental group received no treatments. For the second session an additional 32 students were drawn from these same classroom. Again four girls and four boys were selected for each group (experimental and control) were selected.

This study used two different versions of the chore-scheduling task. The first task was identical to the previous study task where the second task was a computer-based chore-scheduling task that was designed to monitor and record each student's performance.

The new task contained the following parts: (1) an 8 1/2 inch by 11 inch colored diagram of classroom, (2) a set of six cards where each card contained a description of one chore (e.g, wiping or throwing away the trash), (3) microcomputers that check their plans interactively, and (4) a graphic interface that allowed the students to see their plan in action. The students could develop their plans by instructing a robot using simple English programming language commands to perform each plan.

The original planning task was given to the experimental group at beginning of the session. After six months those students received Logo programming instruction. At the end of second session, all students were given the new planning task test. The between-participant grouping variables were: 1) Group (Logo, no-Logo) 2) Condition (feedback, no-feedback) and 3) Age (younger, older). For the pretest, the key within-participant variables were the same as those in study one. The key within-participant variables for the second session were: 1) total time for the robot to carry out each of the three plans, 2) total time for thinking about what move to make next in each of the three plans; 3) use of feedback and debugging aids in each of the three plans and 4) degree of similarity of each plan to the others.

Each child was tested individually and this testing procedure is explained in previous study. To determine comparability between both groups, ANOVA was conducted with Age (older,younger) and Groups (Logo, non-Logo) independent variables. ANOVA revealed a significant main effect for the plan (p<.000), but not for the group or the age.

The study hypothesized that students who learned programming will differ from those who did not learn programming in the following ways: (1) programmers spend less time in planning than non-programmers, (2) programmers utilize the available feedback more often and see a listing of their first plans; (3) programmers spend more time early in their plan thinking over alternative plans, and (4) programmers improve their first plan through successive approach each time.

A repeated measure ANOVA was conducted with total time as the dependent variable. Group (Logo, non-Logo), Condition (feedback, non-feedback), and Age (younger,older),were entered as between-subject factors, and plan (first, second, third) as a within-subjects factor. The ANOVA analyses showed that there was a strong main effect for age (p<.000); older students produced better plans overall than younger students. There was also a significant main effect for plan (p<.000). Post-hoc using Newman-Keuls was performed. The analysis showed a significant difference between first plan and both the second and third plans, but between the second and third plans did not differ significantly.

ANOVA was performed for: (1) the number of times students checked the list of chores to be done and (2) the number of times they asked to see listings of their plans. Independent factors were: condition, age, and group. The ANOVA analyses showed no significant main effects on interaction for their variables.

ANOVA was conducted on the total thinking time in plans; the only significant effect was the main effect for plan (p<.000). The Newman-Keuls test showed that students thought significantly more during the first plan than in their second or third plans. However, there was no significant difference in the amount of time that spent thinking in their second compared to third plans. Newman-Keuls post-hoc comparison showed that the first plan was different from the other two, but the second and the third plans were not different from each other.

The finding of the study revealed that learning to program did not differentiate experimental from control group performance. In another words students who learned programming do not preform better in planning skills than students who did not learned programming.

A repeated measured of ANOVA showed that there were no significant effects for group conditions or plans. Thus, there was no evidence to support the claim that the programmers were more likely to follow a model of plan by successive refinement than were non-programmers.

Videotaped sessions were transcribed; however, the interrator agreement was not mentioned. Moreover, the study did not mention the statistical method that was used to analyze the results that were derived from the videotapes. There were neither validity nor reliability reported for the planning task test which was for the pretest and posttest.

The study used the ANOVA to analyze the data; however, all the results of the ANOVA were reported without indicating the unit of analysis or F values. More, ANCOVA rather than ANOVA needed to be performed.

Conclusion

Half of the 21 studies reviewed in this paper showed learning computer programming did not effect students' cognition, and the other half showed some positive effects.

Whether the outcomes of these empirical studies were encouraging or not, it is important to consider whether these results were trustworthy. All studies exhibited one or more of the following problems:

1. Small samples, not randomly selected or assigned; insufficient information about the selection of the sample and socio-economic status of the subjects.

2. There were not enough details about the instruments; the reliability and validity were not reported, and the interrator agreement also was not mentioned; a few of these studies presented face, content or concurrent validity.

3. Incomplete or incorrect statistical results were reported; several studies reportedly showed significant results, but statistical values, unit of analysis, alpha level, or the means and standard deviations were not reported; the number of students used as the unit of analysis when classes were selected, increased the chance of getting type I error; several studies found significant results but the differences between the means were low, and therefore, the result was not of practical value; some of these studies found significant correlations or regressions coefficients, but the r^2 or R^2 was low and again the result was not of practical value; some studies performed more than 10 statistical tests, increasing the chance to get a type I error.

4. Teachers that participated in these studies did not have adequate experience in teaching a programming language; not enough microcomputers were available for students to access; and the length of the treatment was relatively short.

In spite of the deficiencies of these studies, there were some that were better than the others. The studies that were reported validity and reliability could be trusted more than those that did not. Among the 21 studies reviewed in this paper, only one study reported the validity and the reliability of its assessments; the study was conducted by Turner and Land (1988).

Turner and Land (1988) found no significant differences between students who studied Logo and students who studied regular math curriculum on their understanding of mathematics concepts and their cognitive development. However, the same study found significant differences between students who learned more Logo and students who learned a minimal amount of Logo in their understanding of mathematics concepts and cognitive development. These results favored students who learned more Logo. It is important to know that only a relationship was established, not cause and effect.

Referring back to the original question, "Do the skills of computer programs transfer to other students' cognition? " based on the review of these studies, the findings did not support that learning computer programming transfers to the students' cognition.

Even though these studies had weaknesses, their findings were consistent in which they had no effect on students' cognition whether they learned computer programming or not. Therefore, according to the results of these studies, it look as if there is no relationship between learning computer programming and students's cognition.

Recommendation

It seems that the research on learning to program and its transference to students' cognition is reaching a dead lock. Since research began on these issues, similar unpromising outcomes have been repeated over and over. Consequently, it would be difficult to obtain convincing results unless the deficiencies of preceding investigations are overcome.

Researchers need to be certain that teachers who are participating in studies have the experience in teaching computer programming. Moreover, researchers need to be confident that students in the studies are learning programming before they expect transfer to occur. It is clear that if students fail to learn Logo in the first place, one should not expect transfer to occur (Pea & Kurland, 1983).

The first step in science is to describe objectively the subjects that concern the science. It is difficult to conduct a study that examines a complex relationship before collecting a basic description of information that is needed to understand the subjects (Borg, 1987). Therefore, in order to examine a new phenomena, we may start studying this phenomena by conducting a descriptive study. Then, based on the results obtained from the descriptive study, we may find important variables that can be analyzed in further research.

Papert (1980) claimed that learning computer programming will accelerate students' cognitive development. Since Papert's claim, several computer educators examined this relationship between learning programming and students' cognition, before collecting the basic descriptive information that was needed to understand the subjects that they intended to study. Perhaps the reason for not getting encouraging results on these issues, was because the researchers jumped directly to the experimental research, to study the relationship between programming and student's cognition, prior to gathering a basic descriptive information on the nature of the subjects thought processes involved in computer programming.

It maybe worthwhile to start the research in this field by conducting a descriptive study on students' thinking patterns, while they are engaged in computer programming. Likewise, the results of the descriptive research may generate a base or hypotheses for further research studies.

References

Becker, H. (1983). School uses of microcomputers. Baltimore, MD: Johns Hopkins, Center for Social Organization of Schools.

Blume, G. W., & Schoem, H. L. (1988). Mathematical problem-solving performance of eight-grade programmers and nonprogrammers. Journal for Research in Mathematics Education, 19(2), 142-156.

Clements, H. D. (1986). Effects of Logo and CAI environment of cognition and creativity. Journal of Educational Psychology, 78(2), 309-318

Clements, D. H., & Gullo, D. F. (1980). Effects of computer programming on young children's cognition. Journal of Educational Psychology, 76(6), 1051-1058.

Clement, A. C. (1986). Analogical Reasoning and Compute programming. Journal of Educational Computing Research, 2(4), 473-486.

Cunco, D. D. (1980). Young children and Turtle graphics programming. Paper presented at the Annual meeting of the American educational Research Association, San francisco, CA.

Dalton, D. W. (1984). *A comparison of the effects of Logo use and teacher-directed problem-solving instruction on the problem-solving skills, achievement, and attitudes of low, average, and high achieving junior high school learners*. Paper presented at the annual convention of the association for educational communication and technology. Las Vegas, NV.

Degelman, D., Free, J. U., Scarlato, M. P, Blackburn, J. M., & Golden, T. (1986). Concept learning in preschool children effects of a short-term Logo experience. *Journal of Education Computing Research*, *2*(2), 199-205.

Feurzeig, W., Horwitz, P., & Nickerson, M. (1981). *Microcomputers in education*, (Report no 4798), Belt, Beranek, and Newman, Cambridge, Massachusetts.

Gallint, J. (1984). A comparison of the effects Logo and CAI learning environment on skills acquisition. *Journal of Educational Computing Research*, *3*(4), 461-477.

Horner, C. M. (1985). The effects of Logo on attributions toward success. *Computer in the Schools*, *2*(2-3), 45-54.

Howell, R. D., Scott, P. B., & Diamond, J. (1987). The effects of instant "Logo" computer language on the cognitive development of very young children. *Journal of Educational Computing Research*, *3*(2), 249-261.

Judi H., Kemp, T., & Hyslop, I. (1987). *Development and evaluation of the "thinking with Logo curriculum*. Alberta Dept Of Eduction. BBB05991

Kurland, D. M., Pea, R. D., & Mawaby, R. (1986). A study of the development of programming ability and thinking skills in high students. Journal of Educational Computing Research, 2(4), 429-458.

Linn, M.C. (1985). The cognitive consequences of programming instruction in classroom. Educational Researcher, 14(2), 14-29.

Linn, M.C., & Dalbey, J. (1985). Cognitive consequences of programming instruction. Instruction, access, and ability. Educational Psychology, 20(4), 191-206.

Macallister, A. (1985). Problem solving and beginning programming. Paper presented at the annual meeting of the American Educational research Association (96th), Chicago, IL.

Markman, E. M. (1977). Realizing that you don't understand: A preliminary investigation. Child Development, 48, 986-992.

McCoy, L. P. 1986). General variable skills, computer programming and mathematics. Paper presented at the annual meeting of the international association for computing in education, New Orleans, LA.

MacGrath, D. (1988). Programming and problem solving: Will two language do it ?, Journal of Educational computing Research, 4(4), 467-487.

Paper, S. (1980). Mindstorm: Children Computers and powerful ideas. New York: Basic Books.

Pea, R.D., & Kurland, D. M. (1983). _One the cognitive effects of learning computer programming_. (Tech-Rep. No. 9) New York:Bank street college of Education, Center for children and Technology.

Pea, R.D., & Kurland, D.M. (1984). _One the cognitive effects of learning computer programming_. (Tech-Rep. No. 16) New York: Bank street college of Education, Center for children and Technology.

Printrich, P. R., Berger, C. F.,& Stemmer, D. M. (1987). Students programming behavior in Pascal course. _Journal of Research in Science Teaching_, _24_(5), 451-466.

Rieber, L. P. (1986). _The effects of Logo on young children_. Paper presented at the annual convention of the association for Educational communication and Technology. Las Vegas, NV.

Swan, K. (1989). _Programming objects to think with: Logo and the teaching and learning of problem solving_. Paper presented at the annual meeting of American Educational Research Association, San Francisco, CA.

Swan, K., & Black, J. B. (1988). _The cross-contextual to non-computer of problem solving strategies from Logo_. Annual meeting of the American Educational Research Association, New Orleans, LA.

Turner, S. S., & Land, M. . (1988). Cognitive effects of a Logo enriched mathematics program for middle school students. Journal of Educational Computing Research. 4(4), 443-451.

Woolflolk, E. A. (1987). Educational psychology (3rd ed.). Prentice Hall, Inc. Englewood Cliffs, New Jersey.

LEARNING TO PROGRAM AND ITS' TRANSFERENCE TO STUDENTS' COGNITION

Aqeel M. Ahmed
University of Bahrain
Computer Science Department

Abstract

The purpose of this library paper is to review, analyze the empirical research on the benefits of learning computer programming on students' cognition abilities. The paper will focus on this question: Do the skills of computer programming transfer to students' cognition? This paper is organized into three sections based on the measurement outcomes of each study. Section one contains studies that investigated the relationship of programming with general cognitive outcomes; seven of these studies are included in this section. Section two, contains studies that analyze the relationship between learning computer programming and problem solving; eight of these studies are also included in this section. Section three combines a variety of studies that examine learning computer programming with specific cognitive skills; six studies are analyzed in this section. Conclusion and recommendation for future studies are included.