1984

Effects of an integrating computer activity on the transfer and retention of a geometry concept

Greg Davis
Iowa State University, davisgr@iastate.edu

Recommended Citation

Davis, Greg, "Effects of an integrating computer activity on the transfer and retention of a geometry concept" (1984). *Retrospective Theses and Dissertations*. 297.
https://lib.dr.iastate.edu/rtd/297

Effects of an integrating computer activity on the transfer and retention of a geometry concept

by

Gregory L. Davis

A Thesis Submitted to the
Graduate Faculty in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF SCIENCE

Department: Professional Studies in Education
Major: Education (Curriculum and Instructional Media)

Iowa State University
Ames, Iowa

1984

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I: INTRODUCTION

Society is becoming more information oriented (Sheingold, Hawkins, & Kurland, 1984; Molnar, 1980). At the base of this information trend are electronic communication and calculating devices. Calculators and computers provide access to information. Society is left with determining how to use this information to solve problems (Kozmetsky, 1980).

Development of problem solving skills has been a concern of mathematics educators throughout this century. The growing demand for good problem solvers along with declining test scores on national assessment of problem solving tasks create a crisis situation. Organizations such as the National Council of Teachers of Mathematics (NCTM) have recognized this situation. _Agenda for Action_ (NCTM, 1980) listed problem solving as the number one priority in mathematics education:

> "The development of problem solving ability should direct the efforts of mathematics educators through the next decade. Performance in problem solving will measure the effectiveness of our personal and national possession of mathematical competence" (p. 1).

There are two prerequisites to effective mathematical problem solving:

1) the concepts required to solve the problems

2) some general heuristics to aid in development of a strategy to solve the problems.

The development of a student's heuristic skills has been a main instructional variable in problem solving curricula (Suydam, 1980). Research on problem solving has focused on the identification and teaching of problem solving heuristics (Fey, 1982; Kantoski, 1982). It has been shown that problem solvers who are skilled in problem solving heuristics are more likely to become successful problem solvers (Kantoski, 1982). However, current cognitive science information processing theory indicates that the cognitive status of the knowledge is also crucial to problem solving performance.

Cognitive psychologists theorize that problem solving involves a search of particular portions of memory (problem space). The search usually follows some organized pattern (Briars, 1982). Vital material must be associated (linked) with the problem space being examined (schema) for successful problem solution (Mayer, 1981).

Storing new information in memory so that it is linked with current material is called meaningful learning (p. 12, herein). Meaningful learning results in better transfer of information from one problem situation to another. Meaningful learning is evaluated in terms of transfer of knowledge.

For meaningful learning to occur each individual learner must link new content to existing cognitive structures (Mayer, 1977). This individualized form of learning allows the student to develop personal representations of new concepts. Often the teacher will impose their own representation on the student. Instead, effective meaningful learning requires the teacher to become a diagnostician, providing guidance if the students representation is conceptually incorrect.

The current economic situation at all levels of education make teacher based individualized instruction impractical. The computer may provide an acceptable alternative source of individualized instruction (Taylor, 1980).

Beginning in the early 1960s several national reports have supported integrating computer assisted instruction (CAI) into the curriculum (Kantoski, 1982). CAI research which studied the feasibility of using the computer in the instructional process concluded that CAI might save time and produce slightly better achievement. However, the high cost of computer usage was a strong deterrent to most schools.

As instructional computing becomes more affordable, schools are purchasing computer technology. A recent national survey indicated that 53% of a United States schools had at least one microcomputer (Becker, 1983). Fiske (1983) reports school computing power is increasing by as much as 85% each year. The question, "Should schools have computers?" has shifted to "How should computers be used in instruction?" (Sheingold et al., 1984).

There currently exist polar philosophies of how computers should be used in instruction (Luehrmann, 1980). One approach is to program the computer to teach the student. The instructional design incorporated in this approach resembles traditional tutorial and drill sequences. Individualization can be achieved through programmed instruction techniques such as self pacing and remedial branching. The other approach involves creating a learning environment where the student teaches the computer. Seymour Papert (1980), is recognized as a leading proponent of this method. In his book Mindstorms, Papert discusses the two philosophies:

> "In many schools today, the phrase 'computer-aided instruction' means making the computer teach the child. One might say the computer is being used to program the child. In my vision, the child programs the computer and, in doing so, both acquires a sense of mastery over a piece of the most modern and powerful technology and establishes an intimate contact with some of the deepest ideas from science, from mathematics, and from the art of intellectual model building" (Papert, 1980, p. 5).

In view of the discussion of problem solving above, the children teach computer approach appears to be an ideal place to learn concepts. The student has a chance to develop personal representations of concepts which should promote stronger transfer of those concepts to a new learning situation.

The purpose of this study was to provide empirical evidence that a "student teach computer" environment produces strong transfer when compared with a "computer teach student" tutorial approach. Before formally presenting the problem, one more component must be developed.

A Taxonomy of Instructional Computing

Studies comparing CAI with other forms of instruction can usually be categorized in terms of the type of CAI being examined. Many authors agree on the following five categories (Coburn, 1982; Kulik, Kulik, & Cohen, 1980):

1) Tutorial
2) Drill and Practice
3) Simulation
4) Programming
5) Computer Managed Instruction (CMI)

As CAI becomes more sophisticated, these five categories are no longer sufficient. Educational software has been developed containing aspects of several categories. The CAI classification criteria in terms of instructional design is fast becoming obsolete.

Taylor (1980) proposed an alternative classification scheme. Using Taylor's criteria, a computer was used as a tutor, a tool, or a tutee. As a tutor the computer presents information and reacts to feedback from the learner. As a tool, the computer performs a function for the user. For example, word processing or database management. As a tutee, the computer is programmed by a learner.

Taylor's approach allowed an instructional computing view based on the learner's association with the computer, rather then software characteristics. However, both Coburn's (1982) and Kulik's et al. (1980) classification and Taylor's classification provided no indication of expected educational outcomes. For example, a simulation or tutee program might provide advanced organizers for one learner, mastery of a specific concept for another learner, and transfer of knowledge for another learner. The educational outcomes of CAI are determined by the state of the learner, not the instructional design characteristics of the CAI.

This hinders research which investigates instructional uses of computers in terms of educational outcomes. What is needed is a classification scheme for the instructional use of computers based upon the educational outcomes achieved by the learners. Thomas and Boysen (1983, 1984) have developed a taxonomy meeting this criteria.

The Thomas and Boysen taxonomy for the instructional use of computers consists of five categories:

1) experiencing- sets the cognitive and affective stage for future meaningful learning
2) informing- provides new information to the learner
3) reinforcing- develops mastery of new information
4) integrating- new material is associated with existing long term memory via meaningful learning
5) utilizing- using the computer as a tool to perform a task

The taxonomy is valuable to instructional computing researchers because each stage has strong ties to cognitive science topics. For example, advance organizer theory (Ausubel, 1978; Mayer, 1979) can be tied to experiencing instructional computing uses. Mastery and retention are major outcomes of informing and reinforcing. Meaningful learning is the prime objective of integrating. Utilization provides the applied level.

The Thomas and Boysen taxonomy provides a cognitive science (p. 11, herein) based framework within which the effectiveness of CAI can be evaluated. The taxonomy helps instructional computer educators to focus on CAI in terms of educational outcomes. The taxonomy removes attention from the mechanical processes of CAI, and promotes deeper evaluation of the cognitive products.

Statement of Problem

The problem of the study was to determine the relationship of reinforcing versus integrating activities in terms of meaningful learning. The main evaluating tools were a measure for transfer of a mathematical concept from initial learning to a new problem situation, and measures for retention of the mathematical concept.

Purpose of the Study

A design goal was to develop a study founded on the following educational needs:

1) problem solving skill development
2) development of instructional use of computer criteria
3) evidence of the educational outcomes of student controlled computer learning environments.

With these factors in mind, the purpose of this study was to provide evidence that a student teach computer environment was more effective then a drill instructional design in relation to promoting meaningful learning. In terms of the Thomas and Boysen taxonomy, an integrating instructional computing activity was compared with a reinforcing computing activity. Both activities taught a mathematical concept. Measures for transfer of the concept to a new learning situation were used to determine the degree of meaningful learning. Measures of short and long term retention were also administered.

Hypotheses

The hypotheses of this study were:

1. There is no significant difference in the performance on a transfer test between students experiencing a computer based reinforcing treatment and students experiencing a computer based integrating treatment.
2. There is no significant difference in the performance on retention tests between students experiencing a computer based reinforcing treatment and students experiencing a computer based integrating treatment.
3. There is no significant difference on the time needed to complete the treatment between students experiencing a computer based reinforcing treatment and students experiencing a computer based integrating treatment.

Limitations of the Study

The study was conducted in view of the following limitations:

1. The concept taught in this study was suitable for secondary level students (grades 7-12). The experimental population consisted of college undergraduates enrolled in a computer literacy course, creating a wide range of mathematics ability in the population.
2. In terms of academic ability, the collegiate population contains few low ability students. Therefore, there will be no data indicating performance of low ability students on this study's measures.
3. The student teach computer exercises in the experimental treatment requires programming competence. The four hours of instruction prior to the experimental treatments may not have developed a suitable level of programming expertise for a portion of the population.

Definitions

advanced organizer:	Introductory material designed to activate existing cognitive structures in order to facilitate assimilation of new information.
cognitive science:	The combination of cognitive psychology, computer science, linguistics, anthropology, and philosophy relative to the concern with how new learning is integrated into pre-existing structures (Pea, 1984).
computer teach student:	An instructional design based on programmed instruction techniques where the computer provides information to the student. In this study, computer teach student involved drill exercises where the student observed the results of supplied programs.
concept:	A specific set of objects, symbols, or events which share

	common characteristics and can be referenced by a particular word or symbol (Tennyson, Tennyson, & Rothen, 1980).
heuristic learning:	Guided student controlled (discovery) learning (Dwyer, 1974).
heuristics:	An algorithmic (step by step) strategy for achieving a goal state.
long term retention:	The ability to recall information nine weeks post-instruction.
meaningful learning:	Viewed as a process in which the learner connects new material with knowledge that already exists in memory (Bransford, 1979).
problem solving:	The process of achieving an identified goal under specific conditions without previous knowledge of solution.
problem space:	The schema developed during problem solving.
schema:	The components of long term

memory activated during learning.

short term retention: The ability to recall information three weeks post-instruction.

student teach computer: An instructional design where the student programs the computer to perform a task.

transfer of knowledge: The ability to apply previously learned information (knowledge) to a related but more complex problem (Hooper, 1982).

treatment: One of two instructional designs used in this study as a dependent variable.

CHAPTER II: LITERATURE REVIEW

This study examines two computer based approaches to teaching a concept. The instructional use of the computer was a dependent variable; in this case integrating verses reinforcing. In view of this, the review of literature was divided into three sections. Section one contains a review of literature dealing with concept learning. Section two is a review of literature comparing instructional computing approaches. The final section develops the cognitive science basis of integrating computer activities relative to problem solving.

Review of Concept Literature

As reported by Staats (1965), early behaviorist studies (Hull, 1920; Kender & Karasik, 1958) viewed concepts as the response producing group of stimulus elements gained from a group of stimulus objects. By providing groups of objects sharing identified elements, learners were able to "abstract" those elements.

However, as Osgood (1953) pointed out, the process of learning concepts was not unlike learning in general. For Osgood, the key to concept learning was not to identify stimulus object groups to produce abstractions, but to determine how abstractions were produced.

This study investigates varying computer environment effects on the learner's internal state. Explanations of internal states (such as the structure of abstractions) is an identifying tribute to cognitive theory. The discussion of concepts in this review will be cognitive based.

Concept has been operationally defined as a specific set of objects, symbols, or events which share common characteristics (critical attributes) and can be referenced by a particular word or symbol (Tennyson, Tennyson, and Rothen, 1980). Meaningful learning occurs when a concept is associated with existing knowledge. Learning concepts in this way is a prerequisite for self-directed learning. As DuBois (1979) states:

> "Learning concepts and rules result in the capability to perform in novel situations in which students are asked to identify a specific instance of the concept and to apply the rule in a specific instance" (p. 338).

Analysis of concept acquisition instructional variables indicate that a statement of the concept definition along with selected examples and nonexamples best facilitate meaningful concept learning (Klausmeier and Feldman, 1975).

According to Carroll (as reported by Tennyson & Park, 1980), and Merrill and Tennyson (1978), a concept should be defined in terms of its critical attributes. Concept definitions best facilitate concept attainment when stated in terms of critical attributes (Tennyson and Park, 1980). The definition removes the need for the learner to generate critical attributes.

The effectiveness of presenting the concept definition has been demonstrated (Johnson & Stranton, 1966; Anderson & Kulhavy, 1972). In the Anderson and Kulhavy study, a group of college students were presented with a definition before concept learning. This group was better at identifying previously un-encountered instances of the concept.

Tennyson and Park (1980) investigated the relationship between concept definition and examples/nonexamples. Results showed examples were not as effective in teaching the concept as examples preceded by a definition of the concept. Frayer (1970) found that fewer examples/nonexamples were needed if a definition was presented prior to presenting the examples.

A definition or examples/nonexamples will not effectively stand alone in concept instruction. As Klausmeier (1976) pointed out, if a definition only is presented, the student may merely memorize a string of verbal associations. And Tennyson (1973) provides empirical evidence that examples are required to ensure classification.

Selection of examples/nonexamples is done in view of relationships between the examples. Three variables in this relationship are: (1) divergency of examples, (2) relative difficulty levels of examples, and (3) similarity of variable attributes among matched examples and nonexamples (Tennyson, Woolley, & Merrill, 1972). Meaningful learning of a concept can be optimized if the example/nonexample relationships are easy-to-difficult, divergent, and examples are matched with nonexamples on the basis of similar variable attributes (Tennyson et al., 1972; Houtz, Moore, & Davis, 1973).

Determining the number of examples is another aspect of concept instructional design discussed in the literature. Early work by Clark (1971) based upon a review of 1960 literature, found the optimal number of examples which can be presented simultaneously to be four.

Markle and Tiemann (1969) proposed that the number of examples should be based on the complexity of the concept (in terms of critical and variable attributes). The search for an absolute number of examples was also discouraged by Klausmeier and Feldman (1975). In general, the more abstract the concept, the more examples needed for learning.

Merrill and Tennyson (1977) have proposed that concepts do not exist in isolation, but as part of a set of related concepts. Often examples for one of these "coordinate concepts" will be a nonexample for another. Tennyson et al. (1980) reported nonclassification of a concept by a learner can most effectively be altered by teaching the discriminating coordinate concepts. They found a computer assisted concept learning environment where the computer presents instruction based upon a model of the student to be more effective than a learner controlled environment.

Tennyson and Park (1980, p. 65) have proposed a four step process for teaching concepts:

1) The taxonomical structure of the content should be determined. The three levels of concept structure--superordinate, coordinate, and subordinate-- should be analyzed with identification of critical and variable attributes.
2) A definition of the concept should be prepared in terms of the critical and variable attributes.
3) The examples should be arranged in rational sets by appropriate manipulation of the attributes. Within a rational set, containing one example

from each coordinate concept, the examples should have similar variable attributes.

4) The presentation order of the rational sets should be arranged according to the divergency and difficulty level among examples of the concept, and the presentation order of the examples within rational sets should be decided according to updated information about the learner's knowledge state.

Tennyson, Youngers, and Suebronthi (1983, p. 280) later modified the process to consist of two learning processes:

1) formation in the memory of information representative of a given concept class
2) development of the cognitive skill to use the representative information in evaluating specific dimensions of similarity and difference between and among newly encountered instances.

For effective concept learning, it seems necessary to provide a concept definition and examples of the concept for the learner.

Review of CAI Literature

The second section of this chapter reviews theoretical and empirical reports dealing with instructional uses of the computer. The focus is on educational outcomes produced by various computer based instructional designs.

At the heart of computer based instructional design is the issue of "computer controlled" versus "student controlled" learning environments. This issue will be developed by first reviewing literature dealing with computer controlled instruction (CAI). Then literature dealing with "student teach computer" environments is presented, including a discussion of its cognitive science base.

CAI first appeared in the late 1950s, with early work centered at Florida State University, Dartmouth, and Stanford (Chambers and Sprecher, 1984). CAI research has centered upon the feasibility of using the computer in the instructional process (Hooper, 1982). In most cases, CAI was based on instructional designs similar to traditional forms of instruction.

When comparing CAI to the traditional instruction it replicates, researchers have based their evaluations on various educational outcomes. CAI effectiveness has been measured in terms of initial learning, retention of learning, time on task, as well as changes in attitude toward computers (Suppes & Morningstar, 1972).

Upon reviewing CAI literature, Chambers and Sprecher (1984, p. 12) compiled four educational characteristics of CAI.

1) The use of CAI either improved learning or showed no differences when compared to the traditional classroom approach.

2) The use of CAI reduced learning time when compared to the regular classroom.

3) The use of CAI improved student attitudes toward the use of computers in the learning situation.

4) The development of CAI courseware following specified guidelines can result in portability and their acceptance and use by other faculty.

Based on a meta-analysis of CAI (Kulik, Kulik, and Cohen, 1980), Kulik (1983) made conclusions similar to Chambers's and Sprecher's. Although the Kulik et al. analysis did include reports on programming, the majority of the data was based on "computer teach student" environments. The meta-analysis included information from fifty-nine CAI studies. These data were grouped in terms of four major applications of the computer to instruction (tutoring, computer managed teaching, simulation, and programming).

The Kulik et al. (1980) study did include child teach computer (programming studies). But the meta-analysis was designed to measure effect size of each of the fifty-nine studies. This indicates the strength of various reported effects, but does not give indication of educational outcomes underlying the effects.

From the Chambers and Sprecher (1984) and Kulik (1983) reports, it seems that CAI has been established as an effective instructional experience. However, the literature has not shown CAI to produce strong integrating outcomes. CAI has not been shown to promote transfer more effectively then the traditional forms of instruction which it replicates.

The apparent ineffectiveness of CAI to promote transfer can be contrasted with the meaningful learning outcomes promoted by student teach computer proponents.

The leading proponent of "student teach computer" learning experiences is Papert (1971a, 1980). Papert has founded his philosophies on a strong cognitive psychological base. Papert envisions a learning environment where the child is in control of the learning process. In this environment, learners work toward pre-determined goals by developing personal representations of their solutions. If goals are well chosen, learners are forced to associate new information with information existing in long term memory. This association provides cognitive links which facilitate transfer. Papert's child teach computer philosophy is closely related to the discovery learning approach

(Bruner, 1966, 1965; Strike, 1975; Cohen, 1975), and directly addresses the need for cognitive based instructional design.

To provide a model of a student teach computer environment, Papert developed LOGO, a high level (user friendly) programming language designed for elementary age students. Using LOGO a "mathland" can be developed where students experience geometry by creating line graphics (turtle geometry). In mathland, students solve turtle geometry based problems. Like other proponents of problem solving skill development through programming (Mayer, 1981), Papert proposed that turtle geometry problem solving would develop a learner's ability to use heuristic knowledge. Additionally, in Papert's view the student controlled nature of LOGO would allow students to "learn to think of formal mathematics as rooted in intuitive-body mathematics" (Papert, 1980).

As noted by Pea (1984), child programming learning outcomes claimed by Papert and friends maintain a solid theoretical basis. What is needed is empirical evidence of the claims.

Review of Literature Related to Integrating

Recall the introduction's discussion of the Thomas and Boysen (1983, 1984) five stage taxonomy of instructional computing. This section will review literature that relates to using integrating stage activities to develop problem solving ability.

Each taxonomy stage develops particular steps in the human information processing system. Figure 1 illustrates several taxonomy/information processing links.

Mayer's (1975) model of information processing is used in this figure. Considerations in Mayer's model are:

1) how much information is received
2) how much prerequisite knowledge the learner has
3) what aspects of the learner's existing knowledge are activated during learning and used as an assimilative set to be integrated with new material.

Relationships of the model to taxonomy stages are indicated. Briefly, experiencing activities are used to make sure critical knowledge exists. Informing and reinforcing activities insure that knowledge is received. Integrating activities broaden the search for existing knowledge which can be linked to the new knowledge.

Based on cognitive theory then, learning a concept via an integrating computer activity rather then a nonintegrating computer activity subjects the concept to a wider variety of pre-existing knowledge. Potentially,

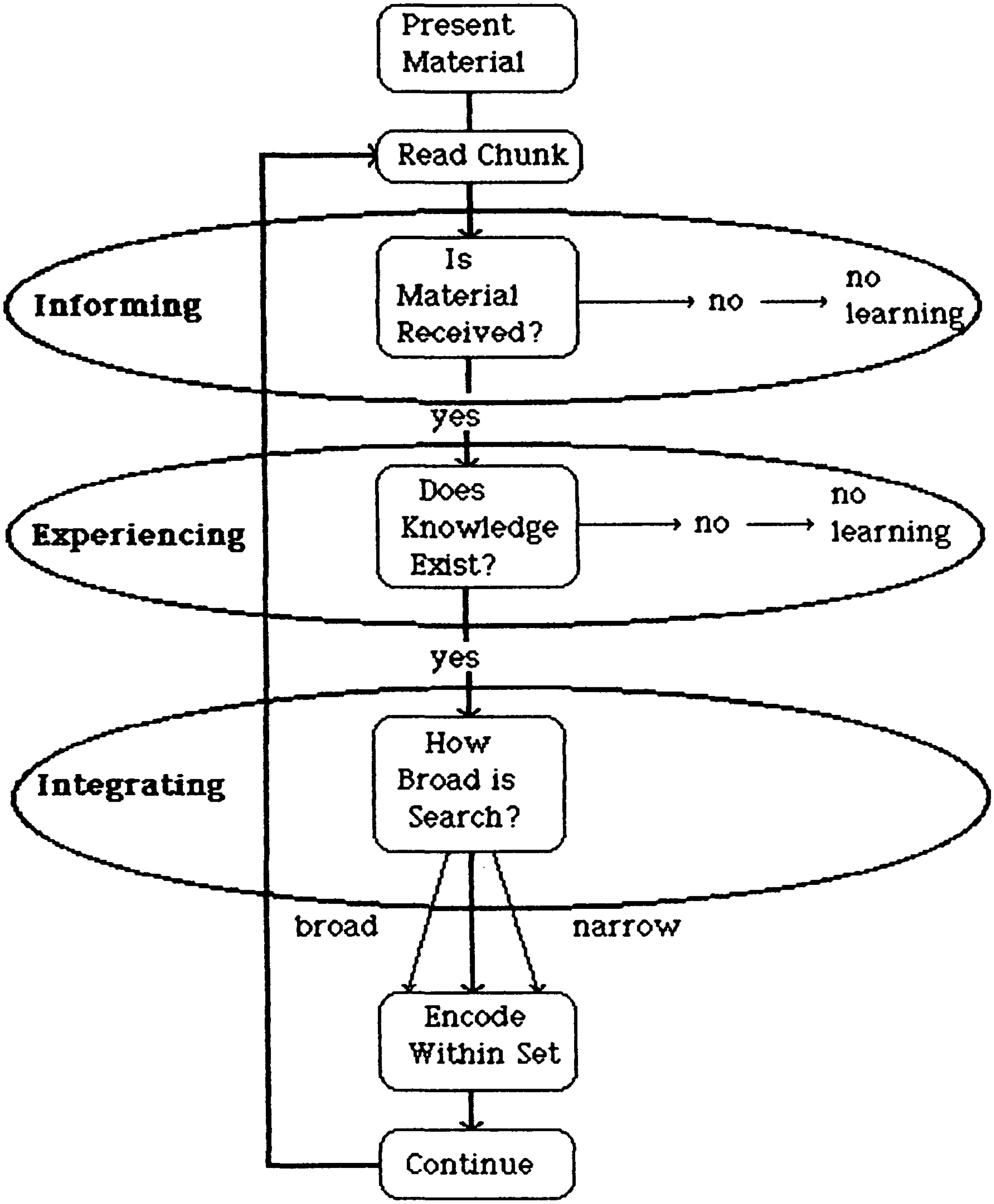


Figure 1. Taxonomy relationships to Mayer's information processing model

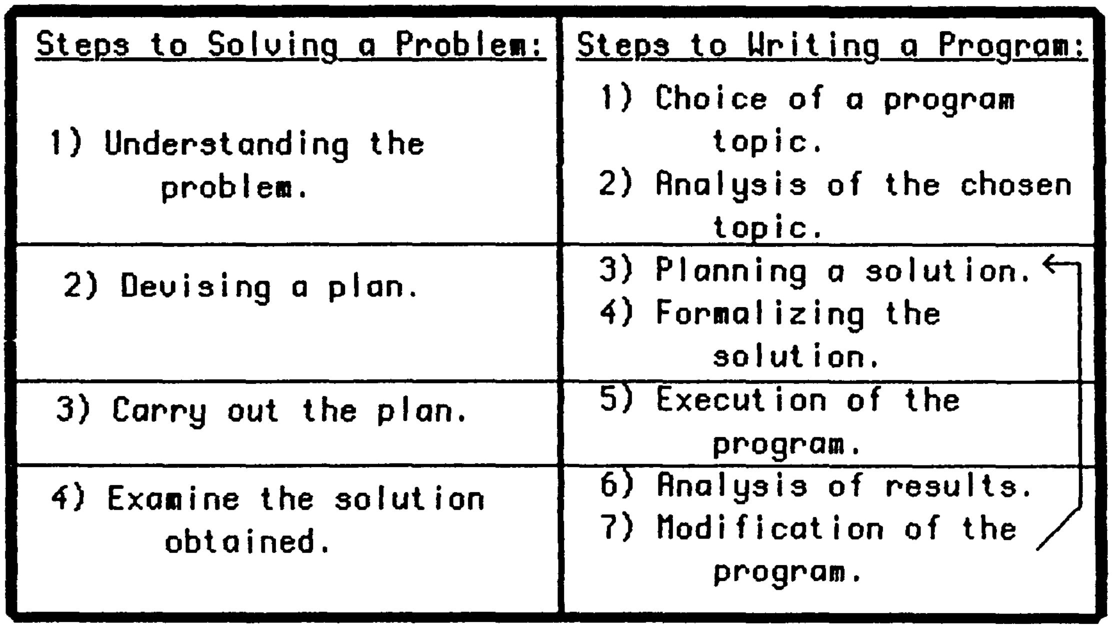
the concept will be linked to more of the pre-existing cognitive structure of the learner.

The main research tool used to measure the effects of integrating type computer activities has been the measure for transfer of knowledge. Transfer of knowledge is influenced by the degree which new information is integrated with a learner's prior knowledge (Dansereau, 1980; Ausubel, 1978). Transfer of principles and attitudes is at the heart of the educational process (Bruner, 1960; Mayer, 1975; Dansereau, 1980). If further research supports integrating activities as a facilitater of transfer, then the integrating instructional use of a computer may replace more traditional CAI (tutorial, drill and practice) as the most used form of instructional computing.

One possible opportunity for integrating computer activities is provided by artificial intelligent computer based instruction. Goldstein (1980) has developed a model for a computer coach. The coach develops a representation of the learner's cognitive status relative to a learning activity. The coach's representation evolves based upon a psychological analysis of the student's performance relative to an expert's performance. The representation controls student performance feedback provided by the coach. Goldstein reports that coaching strategies develop problem solving abilities by (1) allowing students to focus on complex problems, (2) providing a simulation environment where consequences of various actions can be explored, and (3) providing a programming environment in which students implement their own problem solving programs.

Goldstein's third point seems to hold the most promise in terms of developing problem solving skills. Not only does programming the computer to perform a function based on a new concept theoretically open the learner's existing cognitive structure to the new concept. The process of programming closely resembles the processes of problem solving.

Figure 2 indicates the relationship of four phases of good problem solving with a list of computer programming stages. There are some similarities. Computer programming consists of a sequence of problem solving processes. The programmer has a goal. To meet the goal, he constructs a program. During this construction, there will often be new problems (bugs) created. To achieve the main goal, a programmer must solve these new bugs. Although the debugging process is often frustrating (and sometime provides obstacles greater than the original problem/goal itself), debugging does provide problem solving practice. A learner becomes an expert in many



Steps to Solving a Problem:	Steps to Writing a Program:
1) Understanding the problem.	1) Choice of a program topic. 2) Analysis of the chosen topic.
2) Devising a plan.	3) Planning a solution. ← 4) Formalizing the solution.
3) Carry out the plan.	5) Execution of the program.
4) Examine the solution obtained.	6) Analysis of results. 7) Modification of the program.

Figure 2. Steps to solving a problem and steps to writing a program

problem solving situations simply through practice (Anderson, 1980).

A current point of debate questions if computer programming problem solving skills transfer to new nonprogramming problem solving situations. Reif (1980) argues that contemporary "structured" programming design was developed after in-depth analysis of human information processing. The resulting "top-down" approach to programming becomes a powerful heuristic which can be used in non-programming problem solving. Consider the following points (Reif, 1980):

(1) Structured programming, although inspired by the existence of computing machines, is really not centrally concerned with computers; rather, it addresses the question of how human beings can effectively solve the problem of writing complex programs (irrespective of whether or not these are ultimately implemented on a computer). Thus, the precepts of structured programming are prescriptive and specifically designed to enhance human problem solving in a particular domain.

2) These precepts have, in practice, been found to be quite successful and are coming to be increasingly used. Indeed, they provide a

generally applicable framework within which one can embed more specific knowledge about particular algorithms or other detailed aspects of computer programming.

3) Some of these general precepts are probably equally applicable in problem-solving domains outside of computer science.

The literature did not contain total support of the view that programming develops problem solving ability. Pea & Kurland (1984) conducted an in-depth analysis of claims similar to Reif's above and reported the following conclusions:

> "We have dismissed the two prevailing myths about learning to program. The myth embodied in most programming instruction--that learning to program is 'learning facts' of programming language semantics and syntax--is untenable for two reasons: (1) it leads to major conceptual misunderstandings, even among adult programmers; and (2) what is taught belies what cognitive studies show good programmers do and know. These studies have direct implications for new content and methods for programming instruction that are under development in several quarters. Studies of learning to program and of transfer outcomes are not yet available for cases where instruction has such nontraditional emphases .... We have also argued against the second myth--the spontaneous transfer of higher cognitive skills from learning to program to other domains. Resistance in learning to spontaneous transfer and the predicted linkages of kinds of transfer beyond programming to the learner's level of programming skill were major points of these critical reviews" (p. 28).

Reguardless of the position, authors consistently called for further empirical research into programming related topics.

CHAPTER III: METHODS

Subjects

One hundred ninety college undergraduates participated in this study as part of an Iowa State University College of Education computer literacy course, Secondary Education 101. Iowa State University is located in central Iowa, and at the beginning of the 1983 academic year had a total enrollment of 26,020 students.

Secondary Education 101 was designed for pre-service education students. However, a wide variety of majors were enrolled in the class. For this experimental population, approximately 45% were from the College of Agriculture, 45% from the College of Education, 5% from the College of Science and Humanities, 3% from the College of Design, 1% from the College of Home Economics, and 1% other.

Students enrolled in Secondary Education 101 attended two one hour lecture sections and one two hour laboratory session each week during a semester. Students were assigned in groups of twenty to the laboratory sections, usually at a time of their choosing. Laboratories were held every weekday, normally in the afternoon or early evening. There was one morning section, on Friday.

Laboratories sessions were conducted in a twenty station Apple II+ computer classroom. In addition, a sixteen station homework laboratory was available for students.

Rather then building computer literacy through one main activity (such as programming), Secondary Education 101 was designed to provide a variety of computer experiences. The subjects were exposed to one week of BASIC programming, two weeks of word processing, three weeks of LOGO programming, one week of database management, two weeks of spreadsheet management, two weeks of CAI design using an authoring language, and two weeks of experiences on a time-sharing system. Past experience with Secondary Education 101 indicated this curricula was effective in increasing student self-assessment of computer ability, while lowering student self-assessment of computer anxiety (Thompson, 1983).

The study itself was conducted four weeks into the course during the three week LOGO experience. Retention measures were included in both the midterm (two weeks post-treatment) and final (nine weeks post-treatment) exams. In accordance with Iowa State University's Human Subjects Committee, informed consent was obtained from each student participating actively in the study.

Instruments

Information from the following instruments was used for data analysis:

1) precourse assessment
2) pre-experimental test (pretest)
3) treatment exposure time
4) post-experimental test (posttest)
5) short term retention test
6) long term retention test

The precourse assessment is given to all Secondary Education 101 students upon entering the course. The survey collected information dealing with previous computer experience, self-assessed computing literacy, and familiarity with computer terminology and operations. For purposes of this study, the survey provided an ordinal ranking of students in relation to incoming computing literacy. A copy of the survey is included in Appendix B.

The pre-experimental test (pretest) was designed to measure entering knowledge of the mathematical concept being studied. Specifically, the pretest identified subjects in the experimental population who already understood the concept being taught. In addition, the pretest measured knowledge of LOGO programming procedures, and obtained the informed consent of active participants. The pretest was designed to be machine scored. Appendix C contains a copy of the pretest.

When the experimental population experienced the treatment, each subject recorded the time spent using the treatment materials. This time was recorded in minutes on the subject's treatment answer sheet. The time was later coded on the pretest answer sheet by an instructor.

The post experimental test (posttest) was divided into two sections. Part one was nearly identical to the pretest. The only alteration was to replace the informed consent question with a question asking for definition of the concept. Part one was used to determine if the state of the learners changed relative to the concept taught. Appendix D contains a copy of part one of the posttest.

Part two was designed to measure short term retention and transfer of the mathematical concept to a new learning situation. It consisted of four hand scored questions. Each solution was evaluated in terms of syntax correctness as well as conceptual correctness. Scoring of the transfer task was patterned after methods used by Schoenfeld (1980) and Hooper (1982). Appendix E contains a copy of the second posttest part.

Short and long term retention tests were included as nongraded parts of the Secondary Education 101 midterm and final exams. Identical questions were used on both exams. A copy of the questions as they appeared on the final exam is included in Appendix F.

Treatment

Two treatments which varied on instructional use of the computer were used to teach a mathematics concept, the Total Turtle Trip Theorem.

In this case, the turtle was the pointer in LOGO graphics. The proposition from Papert (1980) of the total turtle trip theorem was:

> "If a Turtle takes a trip around the boundary of any area and ends up in the state which it started, then the sum of all turns will be 360 degrees (count right turns as positive, left turns as negative)" (p. 76).

The Total Turtle Trip Theorem was chosen as an appropriate concept based on its relationship to geometry, its ability to be expressed in LOGO turtle geometry, and the capability to apply it to a convenient transfer situation. In addition, presenting the total turtle trip treatments did not interfere with Secondary Education 101's instructional goal of providing LOGO programming experiences.

Both treatments utilized cognitive psychology based approaches to teaching concepts. In both cases, a concept definition was provided, followed by examples containing concept attributes. The treatments' main dependent variable was the instructional use of the computer in presenting the concept examples.

Treatment One (control group) subjects experienced a tutorial and drill instructional design. After reading the definition, control group subjects were provided with LOGO code, asked to enter the code, and observe the results. In terms of the Thomas and Boysen (1983, 1984) taxonomy, these students were using the computer to reinforce the concept (computer teach the student). See Appendix G for a copy of Treatment One.

Treatment Two (experimental group) subjects were also presented with the concept definition. Then, they were asked to program the computer using LOGO to create specific regular polygons. For students who based their solutions on the Total Turtle Trip Theorem, this was an integrating activity. See Appendix H for a copy of Treatment Two.

Uniform written instructions for the treatment activities were provided to both groups. Instructors for the control group were present, but were only allowed to answer questions dealing with LOGO programming. The experimental group had one instructor who provided programming information. This instructor also provided feedback to students with inaccurate representations of the concept.

In summary, two experimental treatments were used to teach a geometry concept, the Total Turtle Trip Theorem. One treatment utilized a computer teach student component. The other treatment used a student teach computer approach.

Research Design

The research design for this study was based on the Campbell and Stanley (1966) pretest-posttest control group design. The content of the treatments was designed to teach a geometrical concept, the Total Turtle Trip Theorem. The instructional use of the computer was varied in each treatment. One treatment was designed as a reinforcing activity. The other treatment was designed as an integrating activity.

Figure 3 graphically displays the classic pretest-posttest design. In this design, the entire experimental population is randomly assigned to control or experimental groups (R), observations of those groups are made (O_1), experimental treatments are given (T_1 and T_2), and finally observations are once again conducted (O_2).

A graphic representation of this study's design is

R	O_1	T_1	O_2
R	O_1	T_2	O_2

R = Randomization
O_1 = Pretest
O_2 = Posttest
T_1 = Control treatment
T_2 = Experimental treatment

Figure 3. Representation of classic Campbell and Stanley (1966) pretest/posttest control group experimental design

O_1	R'	O_2	T_1	O_3	O_4	O_5
O_1	R'	O_2	T_2	O_3	O_4	O_5

O_1 = Precourse assessment
O_2 = Pretest
O_3 = Posttest
O_4 = Midterm retention test
O_5 = Final retention test
R' = Stratified randomization
T_1 = Control treatment
T_2 = Experimental treatment

Figure 4. Representation of design used for this study

shown in Figure 4. This design differs from the Campbell and Stanley design in several ways. First, an observation is made of the entire class (O_1, precourse assessment) at the beginning of the course. Next, the control group and experimental groups are chosen at random within each preassigned laboratory section (R'). This method of stratified randomization was selected because randomization of the entire experimental population without regard to laboratory meeting time could have produced a laboratory meeting time bias, i.e. a certain type of student might choose a Friday morning lab, and total group randomization might place all of these "special" students in one treatment group.

Upon randomization, each group was given a pretest (O_2 followed by an instructional treatment (T_1 or T_2). Both experimental and control groups received a treatment. The treatment which included integrating computer activities was designated as the experimental treatment. After completion of the treatment, both groups were given instruction on material to be used in the transfer measurement. The transfer measurement was conducted the following week as part of the posttest (O_3). At later dates, measurements of retention were conducted (O_4 and O_5).

Research Procedures

In view of the inherent generalizability limitations of the study, it was conceded that the college level laboratory setting would provide reduced external validity in relation to secondary school mathematics. However, before conducting classroom based research aimed at generalizability, it was important to first empirically support the hypotheses of this study. Therefore, emphasis was placed on internal validity factors during the development of the following design procedures.

Because the precourse assessment and long term retention test provided data, the study effectively was conducted over the entire fifteen weeks of spring semester. However, the pretest, posttest, and treatments were conducted over a span of three weeks, beginning with the fourth week of the semester.

The first aspect of the procedure to be considered was the initial development of student LOGO programming skills. Students in the experimental group would require skills sufficient to program the Total Turtle Trip based exercises into the computer. Although students in the control group would not require this programming ability, selection-maturation interaction effects on internal validity as well as Secondary Education 101 course requirements suggested that the entire population be given programming instruction.

This programming development was accomplished in the first LOGO laboratory session. Students practiced writing small LOGO programs, and were given a programming assignment to complete as homework. A copy of the assignment is included as Appendix J. The laboratory experience was supplemented with instruction and examples in two lecture sections.

During the second week at the time of each of the ten laboratories, the pretest and treatments were given. The pretest obtained informed consent, measured basic LOGO programming ability, and measured knowledge of the Total Turtle Trip concept. Included in the informed consent procedure was a description of the project and notification that study performance and effects would not be included in student course evaluation. The test for programming ability indicated lack of programming skill for any particular participant at time of treatment. Data from subjects with inadequate programming ability were not included in the study. The test for pre-treatment concept knowledge indicated which students already understood the concept to be taught. Data from these students were not included in the study.

Upon completion of the pretest, control and experimental groups were given the experimental treatment. Both groups were allowed up to one hour to complete the activity. At the time of the pretest and treatment, control groups and experimental groups were placed in separate computer classrooms to guard against control/experimental group interactions. All experimental groups worked with the same instructor. The instructor for the control groups varied. However, the self-contained nature of the control group treatment allowed the instructor to serve as a monitor only, reducing the threat to validity.

The second hour of every laboratory was conducted by the same instructor. The hour was devoted to advanced LOGO programming techniques using variables. During the second hour, both experimental and control groups were brought together.

During the third week of the experiment, within their assigned laboratory sections, all subjects were given the posttest. The posttest measured retention of the concept, as well as transfer of the concept to a new learning situation. The transfer activities consisted of asking questions that required knowledge of the Total Turtle Trip Theorem and knowledge of LOGO variables.

The transfer activity was evaluated using a classification scoring system based on suggestions by Schoenfeld (1982) and resembling methods used by Hooper (1982). Based on pre-determined criteria, each subject's answer was analyzed and placed in one of four categories: Incorrect, partial credit, syntax errors, and correct.

Figure 5 shows a correct solution for the transfer task. Subjects with equivalent answers were placed in the correct category. If a subject had an equivalent answer with exception of LOGO syntax errors (missing colon, missing bracket, etc.), they were placed in the syntax errors category.

The circled items in Figure 5 were determined to be key parts to the solution. If an answer lacked exactly one of these key parts, the subject was placed in the partial credit category. If more then one part was missing, the subject was placed in the incorrect category.

The transfer evaluation was conducted by the researcher as well as independent scorers. When few discrepancies (less than 5) were found, the researcher's data were excepted as unbiased and used in the statistical analysis.

Following the posttest, two other measures were conducted. A measure of concept retention was included on the Secondary Education 101 midterm and final exams.

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

Figure 5. Transfer task solution with key aspects circled

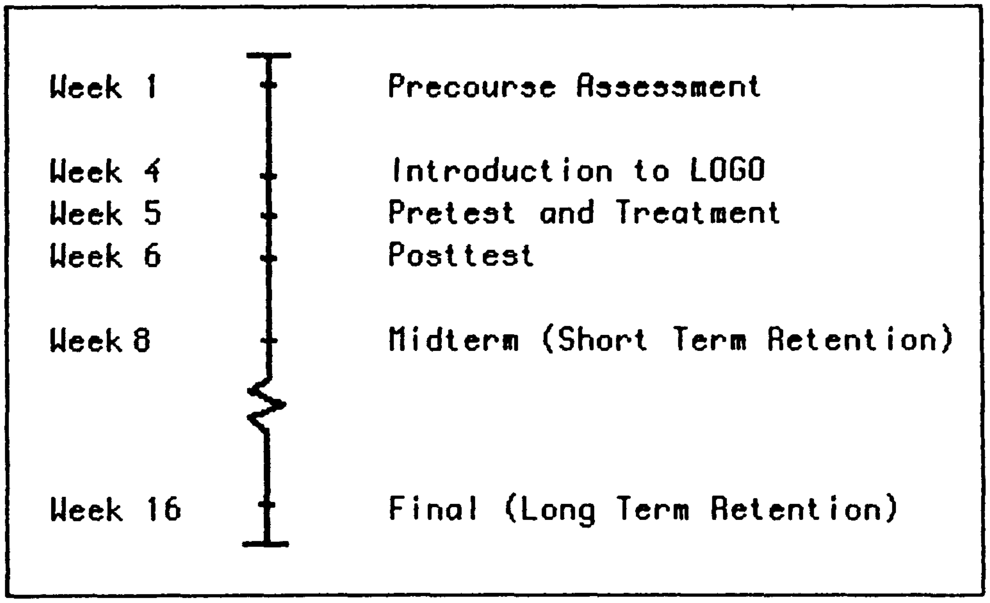


Figure 6. Experimental design real-time relationship

Figure 6 presents a model of the research procedures. Included with the model is a weekly time line.

CHAPTER IV: RESULTS AND DISCUSSION

Results of this study are reported in the first part of this chapter as they relate to each hypothesis listed in Chapter I. This report is followed by a discussion of the findings.

Testing the Hypotheses

As noted in the limitation section of Chapter I, intervening variables associated with the experimental population were identified. These were the subjects' previous mathematical experience, high ability nature, and programming expertise. Pre-experimental knowledge of the Total Turtle Trip concept would render useless post-experimental analysis of the learning task. Lack of computer experience could produce interference with computer based learning tasks. The high ability limitation provided a bias which hindered generalizability.

It was not possible to eliminate the high ability bias of the experimental population. To reduce contamination from the remaining two intervening variables, subjects were measured at time of pretest for basic LOGO skill and knowledge of the Total Turtle Trip concept. The screening criteria resulting from these measures resulted in eliminating 76 cases from the study (from N=169 to N=93).

To determine if the resulting experimental and control groups were equal in terms of computing ability, data from 10 precourse assessment computer self-assessment questions was analyzed. Frequencies of the control and experimental groups are listed in Table 1. Categories specific to this study were general information, present ability, computer anxiety, and computer talent. Chi-square analysis of the combined frequencies of these four categories indicated no significant difference ($\chi^2(4, N=92) = 3.3404$, $p < .5$). It was concluded that the experimental and control groups were equivalent in terms of computer ability prior to the treatments.

Hypothesis 1: There is no significant difference in the performance on a transfer test between students experiencing a computer based reinforcing treatment and students experiencing a computer based integrating treatment.

A chi-square test of independence was used to determine if transfer activity outcomes were independent of experimental treatment. Table 2 contains the frequency of transfer outcomes for each treatment group.

Table 1

Distribution of Experimental Population by
Self-assessed Computing Knowledge

	Rating				
Category	Poor	Lo	Medium	Good	High
General Interest					
Control[a]	0	4	17	12	8
Experimental[b]	3	1	21	10	6
Present Ability					
Control	16	16	9	0	0
Experimental	15	12	14	0	0
Computers as a/ Hobby					
Control	5	11	16	5	4
Experimental	10	10	16	4	1
Computers as/ Appliances					
Control	1	6	16	12	6
Experimental	3	7	10	13	8
Computers and/ General Education					
Control	0	0	6	14	21
Experimental	0	1	5	14	21
Computers in/ Business					
Control	0	0	5	14	22
Experimental	0	0	7	12	22
Computer Anxiety					
Control	2	5	18	8	8
Experimental	3	5	16	14	3
Computer Talent					
Control	11	10	20	0	0
Experimental	10	13	15	3	0
Computers and/ Job-seeking					
Control	1	0	7	16	17
Experimental	1	1	8	18	13
Computers and/ Society					
Control	0	0	3	10	28
Experimental	0	0	4	9	28

Note. The values represent the frequency of ratings for experimental population groups.

[a] Control Group *N* = 41.

[b] Experimental Group *N* = 41.

Table 2

Distribution of Experimental Population on the Transfer Activity

Group	Score			
	Incorrect	Partial Credit	Syntax Errors	Correct
Control[a]	20	9	7	11
Experimental[b]	9	7	8	21

Note. The values represent the frequency of the experimental population on the transfer test.

χ^2 (3, N=92)=7.57418, $p < 0.0557$.

[a] Control Group N= 47.

[b] Experimental Group N= 42.

This distribution approaches significance at the .05 level (χ^2 (3, N=92)=7.57418, $p < 0.0557$). Close examination of the frequencies indicate that the chi-square value is diluted by even distributions in the partial credit and syntax errors only categories. Based on the closeness to .05 significance combined with the observed polarization of the transfer outcomes, Hypothesis 1 was rejected. Transfer outcomes were dependent on treatment type. The integrating treatment produced better performance on the transfer activity.

Hypothesis 2: There is no significant difference in the performance on retention tests between students experiencing a computer based reinforcing treatment and students experiencing a computer based integrating treatment.

There were seven post-experiment retention measurements. Three came at time of posttest, two at time of the Secondary Education 101 course midterm (three weeks post treatment), and two at time of the course final (twelve weeks post treatment). Table 3 contains the frequencies for each individual measure. Data were reported only for subjects completing all phases of the study (N = 58). Chi-square analysis showed no significant difference, indicating that retention of the

Table 3

Distribution of Experimental Population on Retention Measures

Measure	Score Incorrect	 Correct
Posttest Question 1		
Control[a]	0	33
Experimental[b]	0	25
Posttest Question 2		
Control	6	27
Experimental	4	21
Posttest Question 3		
Control	12	21
Experimental	10	15
Midterm Question 1		
Control	9	24
Experimental	10	15
Midterm Question 2		
Control	3	30
Experimental	2	23
Final Question 1		
Control	7	26
Experimental	9	16
Final Question 2		
Control	4	29
Experimental	1	24

Note. The values represent the frequency of the experimental population on short term and long term retention questions.

[a] Control Group *N*= 33.

[b] Experimental Group *N*= 25.

Table 4

Distribution of Experimental Population by Time on Treatment

	Time (minutes)				
Group	10	10-15	15-20	20-25	25
Control[a]	2	11	22	6	7
Experimental[b]	5	7	16	8	9

Note. The values represent the frequency of the experimental population on treatment time groups.

[a] Control Group *N*= 48.

[b] Experimental Group *N*= 45 .

concept was independent of the treatment. Therefore, Hypothesis 2 was not rejected.

Hypothesis 3: There is no significant difference on the time needed to complete the treatment between students experiencing a computer based reinforcing treatment and students experiencing a computer based integrating treatment.

Time in minutes on the treatment task was divided into five groups. Table 4 contains the frequencies of each time group for each treatment group. No significant difference was indicated by the frequencies. This indicates that time on treatment was independent of treatment type. Hypothesis 3 was not rejected.

Discussion

Mathematical problem solving is facilitated by two abilities; to transfer concepts and to apply appropriate heuristics. Most authors agree practice is the key element in heuristic competence. However, the ability to retain concepts (the traditional measurement for concept learning) does not ensure the ability to transfer those concepts.

The purpose of this study was to provide empirical evidence of the educational outcomes of contrasting instructional computing philosophies. A treatment requiring programming and a treatment based on drill were designed. The programming (experimental) treatment used the computer in a tutee (Taylor, 1980) role. The drill (control) treatment used the computer in a tutor role.

Papert (1972b) and Feurzeig (1969) maintained that the programming treatment facilitates problem solving skill development because the process of programming provides practice in the problem solving heuristic of induction. The general hypothesis of this study was that programming the computer also facilitates meaningful learning of the concepts used in the programming problem.

Meaningful learning was operationally defined as the process in which the learner connects new material with knowledge that already exists in memory (Bransford, 1979). Meaningful learning cannot be evaluated by retention measurements. Meaningful learning is measured using test for transfer of knowledge. Information used successfully in a new learning situation indicates the occurrence of meaningful learning.

Instructional uses of the computer that facilitate meaningful learning fall into the experiencing and integrating categories of the Thomas and Boysen (1983, 1984) taxonomy of instructional uses of computers. The key difference between experiencing and integrating activities is the state of the learner relative to the information being presented. In an experiencing activity the learner is presented unfamiliar, general information. The intent is to activate relevant portions of long term memory for later association of new information. An integrating activity is conducted after the learner is familiar with the new information. The intent is to require the learner to actively associate the new information with existing long term memory.

Proponents of child teach computer environments (Papert, 1980; Dwyer, 1974) argued that the learner is required to be active. It followed that programming the computer (a form of child teach computer) to perform tasks based on a specific mathematical concept would constitute a suitable integrating instructional use of the computer.

The experimental population was randomly divided into two groups. Both groups were taught a mathematical concept using microcomputer based instruction. The group designated as experimental completed a four step integrating via programming treatment. The control group completed the same activities via a tutorial and drill treatment. The time spent on the treatment was recorded for each subject. Following the treatments, both groups were given new information which would provide the basis for a subsequent transfer task. Subjects were measured for transfer of the mathematics concept, as well as for short and long term retention of the concept.

Solutions obtained from each subject for the transfer task were categorized. Solutions were scored as totally incorrect, partially correct, totally correct with the exception of syntax errors, or totally correct. The percentage of experimental and control subjects for each category is graphically depicted in Figure 7. Most control subjects were placed in the totally incorrect group, while most experimental subjects were placed in the totally correct group. Chi-square analysis showed this data approaching significance at the .05 level. For this transfer activity, the integrating treatment did appear to facilitate meaningful learning.

An interesting result of this data analysis was the syntax error only category. Because control group subjects were provided examples of syntax correct programs, it was anticipated that they would produce fewer syntax errors than the experimental group. Although this was the case, relative to the incorrect and correct categories the difference was small. This indicates that syntax was not a factor in the transfer activity.

Five questions were used to measure short term

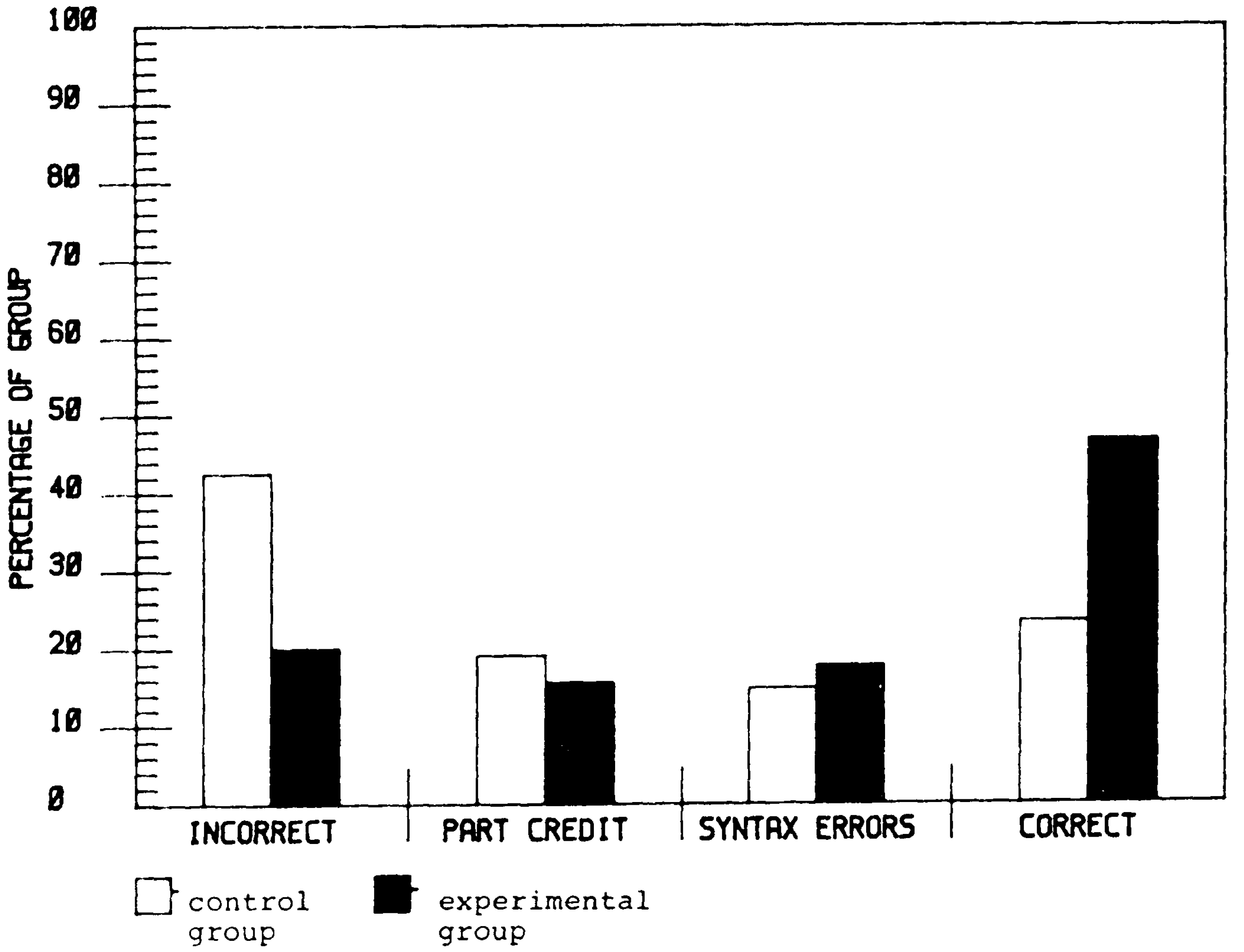


Figure 7. Percentage of each group in transfer activity categories

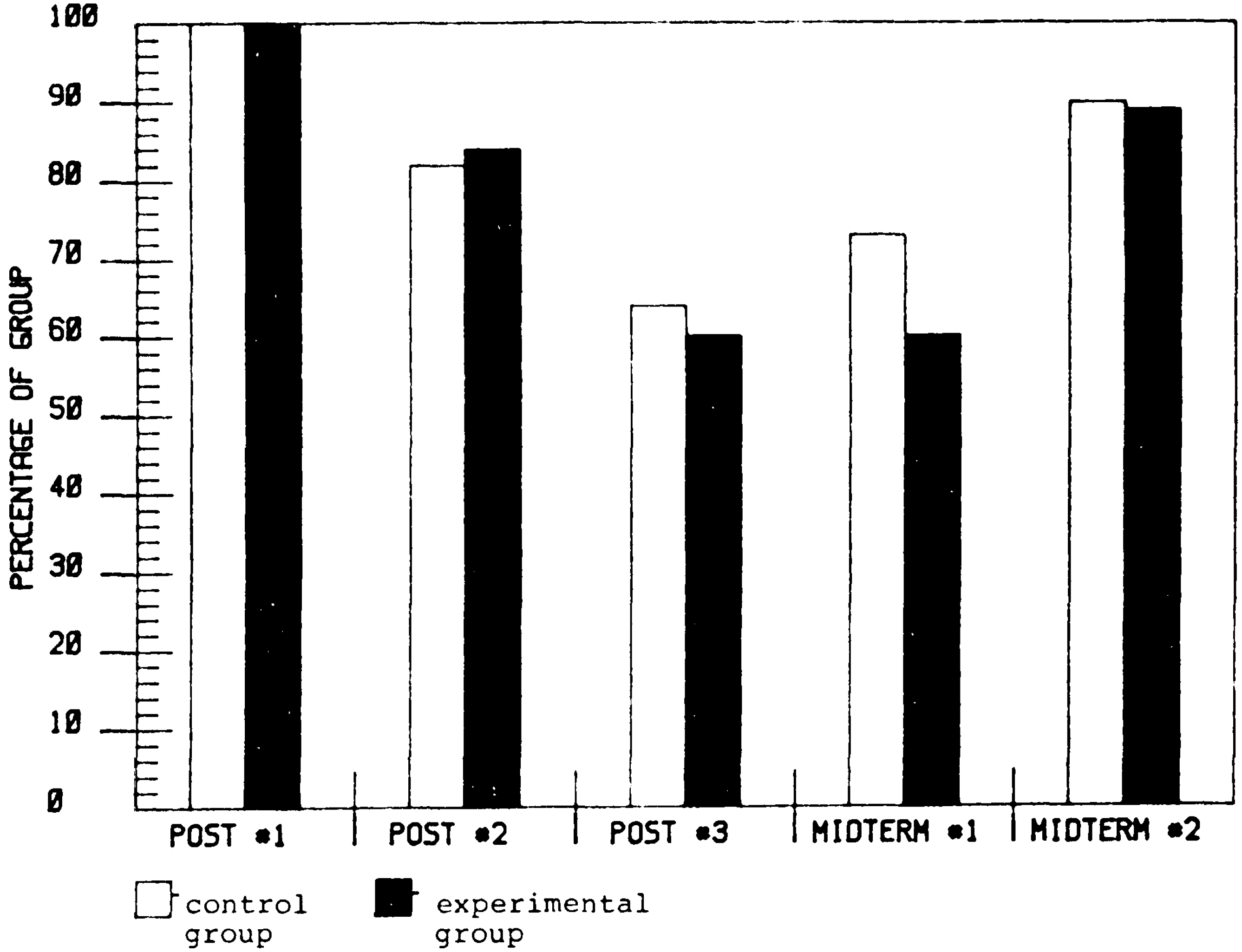


Figure 8. Percentage of each group scoring correctly on short term retention questions

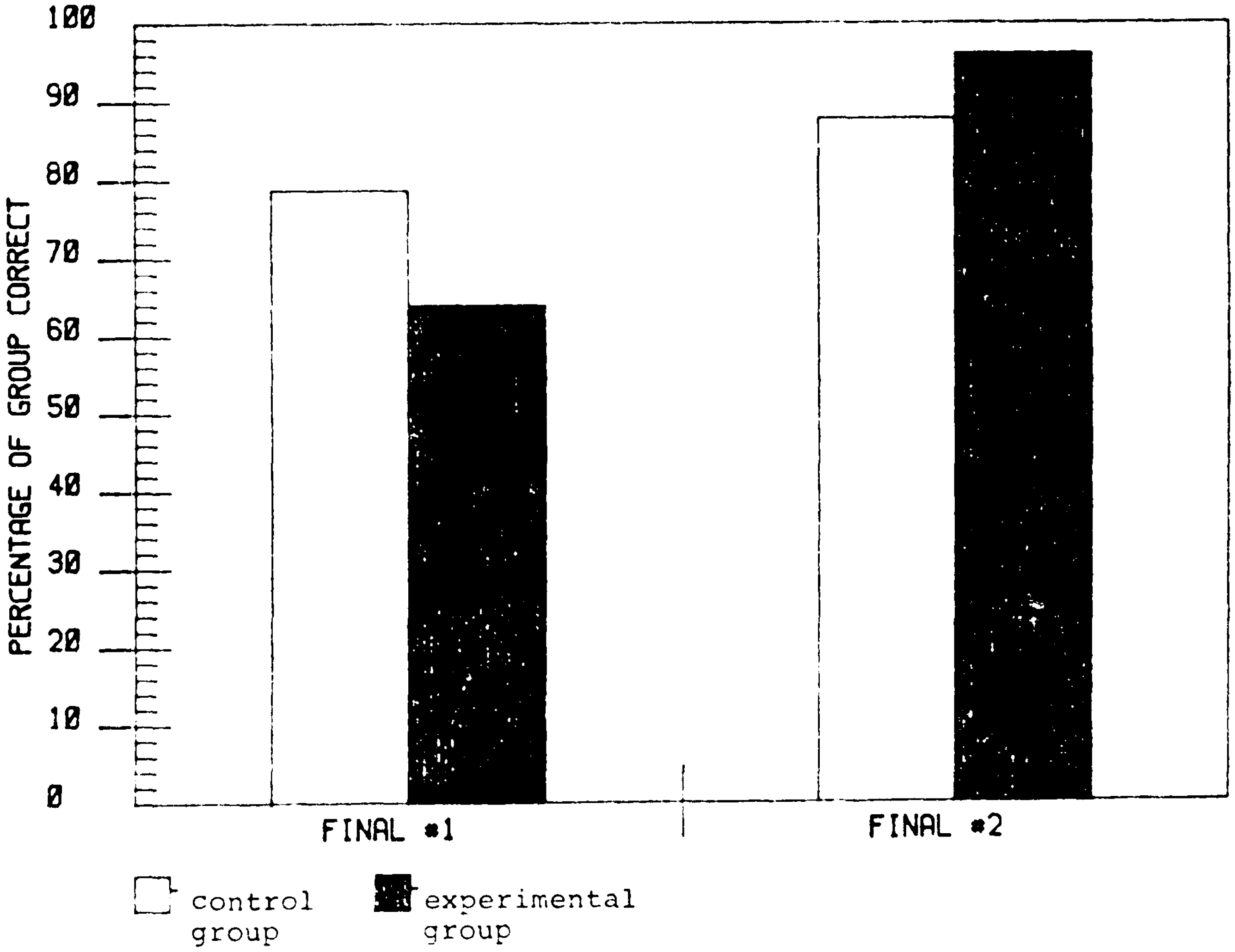


Figure 9. Percentage of each group scoring correctly on long term retention questions

retention. Figure 8 is a graphical representation of the percentage correct on each question for both groups. Although the control group performed slightly better overall on the retention tasks, no significant results were produced. This was somewhat surprising considering that the primary theoretical strength of the control treatment was retention. On these measures, the integrating activity supported retention equally as well as the reinforcing activity. It also can be noted that the experimental group's success on the transfer activity cannot be attributed to a superior retention of the concept.

Figure 9 is a graphical representation of the long term retention results. The two items were included on a final exam given to all subjects. The items were identical to the two midterm short term retention items. No significant difference was found on the long term items. However, it is interesting to note that on one item, 95% of the control group scored correctly on the midterm, while 87% of the control group scored correctly on the final, indicating a drop in retention for the control group. The experimental group showed no drop in retention. Theoretically, the control group's retention performance should decrease because associations that aid in later recall were not developed. The meaningful learning produced by the experimental treatment did produce associations, and long term retention was facilitated.

These findings generally support the theoretical outcomes suggested by child teach computer proponents. The integrating treatment produced better transfer of a mathematical concept, and retention of the concept between groups was not significantly different. Data were also collected to determine the extent of proposed weaknesses of child teach computer environments.

The requirement for increased teacher training and increased teacher involvement is a main criticism of child teach computer environments. Critics suggest these needs fail to capitalize on the computer's special capacity for individualized instruction.

This study did not formally investigate classroom management aspects of the two treatments. Nonclinical interviews with the instructors indicated no difficulty with either treatment. However, the instructors for the child teach computer environment did need to be skilled in the programming language being used. This lends support to the criticism noted above.

The Kulik et al. (1980) meta-analysis indicated that reduced time on task is a main strength of CAI. This study recorded time on treatment for each subject. Based on the Kulik report it was expected that control group subjects would spend less time on their treatment. Figure 10 is a graphical representation of the data obtained. Times were placed into categories. The mean for each group based on category frequencies indicated that the average student in both groups required 15-20 minutes on the treatment. In general, no difference in time on treatment was found between the two groups.

In summary of this discussion, data analysis indicated that a child teach computer treatment produced better performance on a transfer task then a computer teach student approach. No difference in short term retention, long term retention, and time on treatment was found.

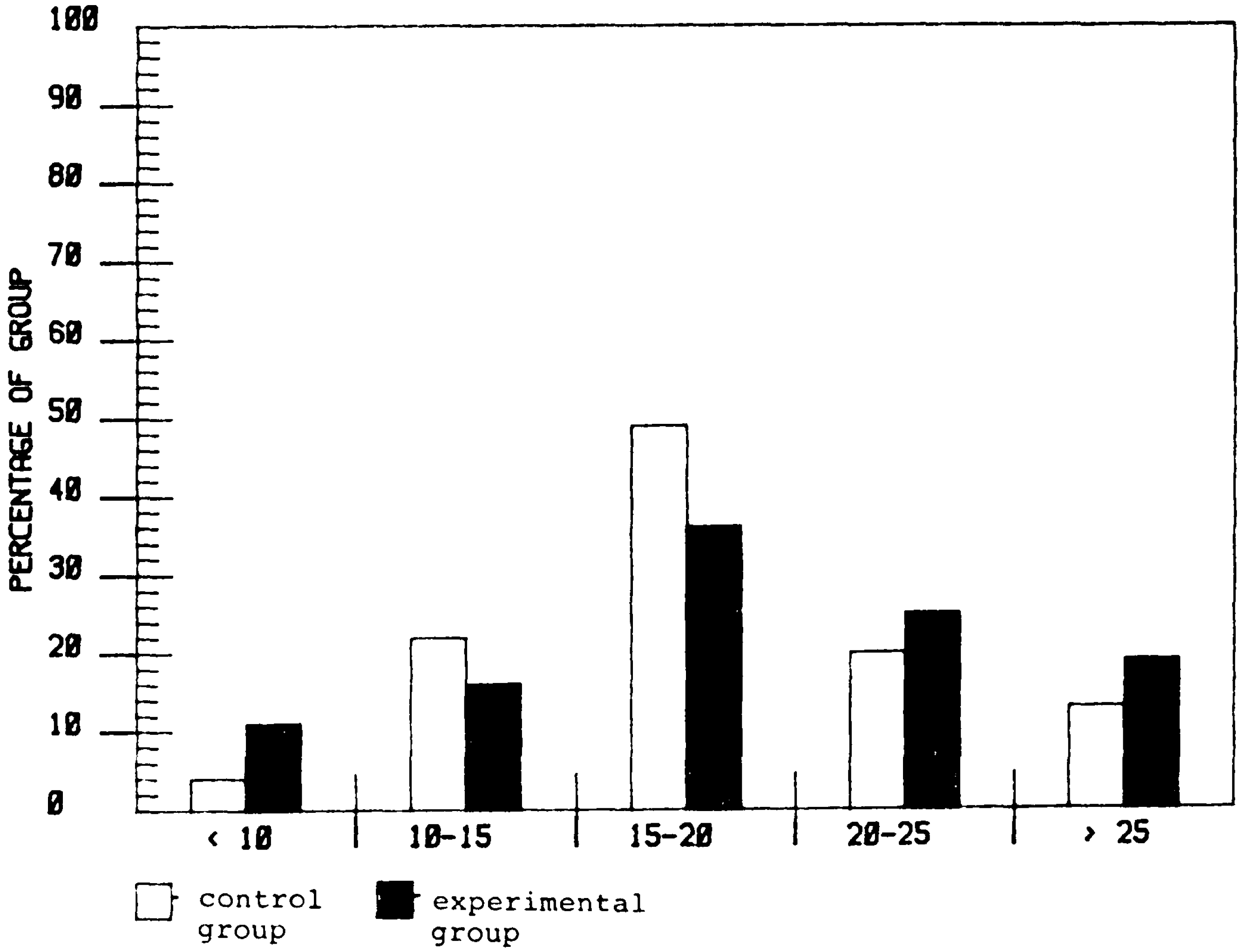


Figure 10. Percentage of each group in treatment time categories

CHAPTER V: SUMMARY, RECOMMENDATIONS, AND CONCLUSION

Summary

The purpose of the study was to compare two instructional computer activities using retention (both short and long term) and transfer measures of a geometry concept, The Total Turtle Trip Theorem. In one activity, the computer was used to reinforce the geometry concept. In the other activity, the computer was used to integrate the geometry concept.

The experimental populaton for this study consisted of undergraduates enrolled in an Iowa State University College of Education course, Secondary Education 101. These students were assigned to one of ten laboratory sections at time of enrollment. These sections were randomly divided into control and experimental groups for this study.

A precourse assessment, pretest, treatment, posttest, and retention tests were given to all students. The only variation between groups was in the treatment. The control group received the reinforcing treatment. The experimental group received an integrating treatment.

Data from population subjects who had insufficient LOGO programming skills, or previous knowledge of the geometry concept taught, were eliminated from data analysis. Three hypotheses were tested using data from the resulting groups.

Hypothesis 1 suggested that there was no difference in transfer of a concept taught with a reinforcing or integrating use of the computer. However, data indicated the integrating activity produced better performance on a transfer activity.

The study looked for a similar difference in performance on retention tasks (Hypothesis 2). No significant differences between groups were found. Investigation of Hypothesis 3, which dealt with time spent on treatment, also produced no significant differences between groups.

Although no significant differences were found, the results from analysis of Hypothesis 2 and 3 were interesting. Advocates of reinforcing computing activities claimed these activities facilitated retention and reduced time on task when compared with noncomputerbased instruction. Data from this experiment indicate that integrating computer activities may be just as effective as retention activities in terms of retention and time on task.

Recommendations

Results on the transfer task provides encouraging support for child teach computer environments. However, experimental population limitations mentioned in the introduction greatly reduce the generalizability of these results. A classroom based experiment would be an appropriate second stage to this study.

Validity of the results could also be strengthend by developing more transfer tasks. Strong claims cannot be based on one measure. Any new tasks should be sensitive to type and degree of transfer. See Dansereau (1980), for a detailed discussion of transfer.

Finally, a true CAI program should be developed for the reinforcing treatment. This would eliminate intervening instructional variables and provide a more formal computer teach student learning environment.

Conclusion

Problem solving skill development is a critical educational need. Child teach computer instruction theoretically provides an effective and exciting environment where these skills can be facilitated. The missing ingredient is empirical research supporting child teach computer claims.

By using a cognitive science base for the study of instruction computing, research will begin to indicate what educational outcomes are produced by specific instructional computing designs. Development of these guidelines will allow educators to select computer activities that produce desired educational outcomes, not replication of old instruction.

REFERENCES

Anderson, J. R. (1980). Cognitive Psychology and its Implications. San Fransisco: W. H. Freeman and Company.

Anderson, R. C. & Kulhavy, R. W. (1972). Learning concepts from definitions. American Educational Research Journal, 9, 385-390.

Ausubel, D. P. (1978). In defense of advanced organizers: a reply to the critics. Review of Educational Research, 48, 251-257.

Becker, H. J. (1983, Fall). Schools uses of microcomputers: Report #1 from a national survey. The Journal of Computers in Mathematics and Science Teaching, 3(1), 29-33.

Bransford, J. D. (1979). Human Cognition. Monterey, CA: Wadsworth.

Briars, D. J. (1982). Implications from information-processing psychology for research on mathematics learning and problem solving. In F. Lester (Ed.), Mathematical Problem Solving: Issues in Research. Philadelphia: The Franklin Institute Press

Bruner, J. S. (1960). The Process of Education. New York:Vintage Books.

Bruner, J. S. (1965). The art of discovery. In R. C. Anderson & D. P. Ausubel (Eds.), Readings in the Psychology of Cognition (pp. 606-620). New York: Holt Rinehart & Winston, Inc.

Bruner, J. S. (1966). Toward a Theory of Instruction. Cambridge, MA:Harvard University Press.

Campbell, D. & Stanley, J. (1966). Experimental and Quasi Experimental Designs for Research. Dallas: Houghton Mifflin Company.

Chambers, J. A. & Sprecher, J. W. (1984). Computer-assisted instruction: Current trends and critical issues. In D. F. Walker & R. D. Hess (Eds.), Instructional Software: Principles and Perspectives for Design and Use (p. 12). Belmont, CA:Wadsworth Publishing Company.

Clark, D. C. (1971). Teaching concepts in the classroom: A set of prescriptions derived from experimental research. *Journal of Educational Psychology Monograph*, *62*, 253-278.

Coburn, P. (1982). *Computers in Education*. Reading, MA:Addison-Wesley Publishing Company.

Cohen, H. A. (1975, May). *The Art of Snaring Dragons* (Artificial Intelligence Memo Number 338). Massachusetts Institute of Technology. (ERIC Document Reproduction Service No. ED 128 179).

Dansereau, D. F. (1980, October). *Transfer of learning from one setting to another*. Paper presented at the meeting of the National Technical Institute for the Deaf.

DuBois, N. F. (1979). *Educational Psychology and Instructional Decisions*. Homewood, IL:The Dorsey Press.

Dwyer, T. A. (1974). Heuristic strategies for using computers to enrich education. *International Journal of Man-machine Studies*, *6*, 137-154.

Feurzeig, W. (1969). *Programming Languages as a Conceptual Framework for Teaching Mathematics* (Report No. 1899). Cambridge, MA: Bolt Beranek and Newman.

Fey, J. T. (1982). Mathematics education. In H. E. Mitzel (Ed.), *Encyclopedia of Educational Research* (5th ed., Vol. 3, pp. 1166-1181). New York: The Free Press

Fiske, E. (1983, August). Computer education: Update '83 part one. *Popular Computing*, *11*, 89.

Frayer, D. A. (1970). *Effects of number of instances and emphasis of relevant attributes values on mastery of geometrical concepts by fourth- and sixth-grade children* (Report No. 116). Madison, WI:Wisconsin Research and Development Center for Cognitive Learning.

Goldstein, I. (1980). Developing a computational representation for problem-solving skills. In D. T. Tuma (Ed.), *Problem Solving and Education* (pp. 53-79). Hillsdale, NJ: Lawrence Erlbaum Associates.

Hooper, E. (1982). *The effects of field dependence and instructional sequence on student learning in a computer-based algebra lesson*. Unpublished master's thesis, Iowa State University, Ames, IA.

Houtz, J. C., Moore, J. W., & Davis, J. K. (1973). Effects of different types of positive and negative instances in learning "nondimensioned" concepts. *Journal of Educational Psychology*, *64*, 206-211.

Johnson, D. M. & Stranton, R. P. (1966). Methods of teaching concepts. *Journal of Educational Psychology*, *57*, 48-53.

Kantoski, M. G. (1982). The use of the microcomputer in instruction for problem solving. In S. Krulik (Ed.), *Problem Solving Concentration Papers* (pp. 59-72). Boston: Allyn and Bacon.

Klausmeier, H. J. (1976). Instructional design and the teaching of concepts. In J. R. Levin & V. L. Allen (Eds.), *Cognitive Learning in Children*. New York: Academic Press.

Klausmeier, H. J. & Feldman, K. V. (1975). Effects of a definition and a varying number of examples and non-examples on concept attainment. *Journal of Educational Psychology*, *67*, 174-178.

Kozmetsky, G. (1980). The significant role of problem solving in education. In D. T. Tuma (Ed.), *Problem solving and education*. Hillsdale, NJ:Lawrence Erlbaum Associates.

Kulik, J. A. (1983). Effects of computer-based teaching on secondary school students. *Journal of Educational Psychology*, *75*, 19-26.

Kulik, J. A., Kulik, C. C., & Cohen, P. (1980, Winter). Effectiveness of computer-based college teaching: A meta-analysis of findings. *Journal of Educational Psychology*, *50*, 525-544.

Luehrmann, A. (1980). Should the computer teach the student, or vice-versa. In R. Taylor (Ed.), *Computers in the School: Tutor, Tool, Tutee* (pp. 129-135). New York:Teachers College Press.

Markle, S. M. & Tiemann, P. W. (1969). *Really*

Understanding Concepts. Champaign, IL:Stipes.

Mayer, R. E. (1975). Information processing variables in learning to solve problems. *Review of Educational Research*, *45*, 525-541.

Mayer, R. E. (1977). The sequence of instruction and the concept of assimilation-to-schema. *Instructional Science*, *6*, 377.

Mayer, R. E. (1979). Can advanced organizers influence meaningful learning? *Review of Educational Research*, *49*, 371-383.

Mayer, R. E. (1981). The psychology of mathematical problem solving. In F. Lester (Ed.), *Mathematical Problem Solving Issues in Research*. Philadelphia: The Franklin Institute Press.

Merrill, M. D. & Tennyson, R. D. (1977, May). Concept teaching: An instructional design guide. *Educational Technology*, *17*, 22-32.

Merrill, M. D. & Tennyson, R. D. (1978). Concept classification and classification errors as a function of relationships between examples and nonexamples. *Improving Human Performance*, *7*, 351-364.

Molnar, A. R. (1980, January). Understanding how to use machines to work smarter in an information society. *Technological Horizons in Education Journal*, *7*, 42-46.

National Council of Teachers of Mathematics. (1980). *Agenda for action*. Reston, VA:Author.

Osgood, C. E. (1953). *Method and Theory in Experimental Psychology*. New York: Oxford University Press.

Papert, S. (1971a). *Teaching Children Thinking* (Artificial Intelligence Memo Number 254). Massachusetts Institute of Tecnnology. (ERIC Document Reproduction Service No. ED 077 241).

Papert, S. (1971b). *Teaching Children to be Mathematicians vs Teaching About Mathematics* (Artificial Intelligence Memo Number 249). Massachusetts Institute of Technology. (ERIC Document Reproduction Service No. ED 077 243).

Papert, S. (1980). *Mindstorms*. New York: N. Y. Basic

Books.

Pea, R. D. & Kurland, D. M. (1984, January). *On the cognitive effects of learning computer programming, a critical look* (Technical Report No. 9). New York: Center for Children and Technology.

Poyla, G. (1954). *How to solve it*. Garden City, NY: Doubleday-Anchor.

Reif, F. (1980). Theoretical and educational concerns with problem solving: Bridging the gaps with human cognitive engineering. In D. T. Tuma (Ed.), *Problem Solving and Education* (p. 46). Hillsdale, NJ: Lawrence Erlbaum Associates.

Schoenfeld, A. H. (1980, January). Measures of problem solving performance and of problem solving instruction. *Journal for Research in Mathematics Education*, 31-50.

Sheingold, K., Hawkins, J., & Kurland, D. M. (1984). *Classroom software for the information age*. New York: Center for Children and Technology.

Siegler, R. S. (1980). When do children learn? The relationship between existing knowledge and learning. *Educational Psychologist*, *15*, 135-150.

Staats, A. W. (1965). Verbal habit-families, concepts, and the operant conditioning of word classes. In R. C. Anderson & D. P. Ausubel (Eds.), *Readings in the Psychology of Cognition* (pp. 18-40). New York: Holt Rinehart & Winston, Inc.

Strike, K. A. (1975, Summer). The logic of learning by discovery. *Review of Educational Research*, *45*, 461-483

Suppes, P. & Morningstar, M. (1972). *Computer-assisted Instruction at Stanford 1966-1968: Data Models and Evaluation of the Arithmetic Programs*. New York: Academic Press.

Suydam, M. (1980). Untangling clues from research on problem solving. In S. Krulik (Ed.), *Problem Solving in School Mathematics 1980 Yearbook of the National Council of Teachers of Mathematics*. Reston, VA: The Council.

Taylor, R. (1980). Introduction. In R. Taylor (Ed.), *Computers in the School: Tutor, Tool, Tutee* (pp. 1-12). New York: Teachers College Press.

Tennyson, C. L., Tennyson, R. D., & Rothen, W. (1980). Content structure and instructional control strategies as design variables in concept acquisition. *Journal of Educational Psychology*, *72*, 499-505.

Tennyson, R. D. (1973). Effect of negative instances in concept acquisition using a verbal learning task. *Journal of Educational Psychology*, *64*, 247-260.

Tennyson, R. D., & Park, O. (1980, Spring). The teaching of concepts: A review of instructional design research literature. *Review of Educational Research*, *50*, 55-70.

Tennyson, R. D., Woolley, F. R., & Merrill, M. D. (1972). Exemplar and non-exemplar variables which produce correct concept classification errors. *Journal of Educational Psychology*, *63*, 144-152.

Tennyson, R. D., Youngers, J., & Suebronthi, P. (1983). Concept learning by children using instructional presentation forms for prototype formation and classification-skill development. *Journal of Educational Psychology*, *75*, 280-291.

Thomas, R. A. & Boysen, J. P. (1983). A taxonomy for the instructional use of computers. Unpublished manuscript, Iowa State University, Computation Center, Ames, IA.

Thomas, R. A. & Boysen, J. P. (1984). The role of computers in professional education. Manuscript submitted for publication.

Thompson, A. D. (1984). A hands-on beginning computer course for pre-service teachers. Manuscript submitted for publication.

Wertz, H. (1981, November). Some ideas on the educational use of computers. *Proceedings of the Association for Computer Machinery 1981 Conference* (p. 101). New York: The Association.

ACKNOWLEDGEMENTS

I note with appreciation the guidance provided by the members of my program of study committee; not only on developing this thesis topic, but throughout my academic career at Iowa State University. These were professors William Rudolph (major professor), Harold Dilts, Anton Netusil, and Jerold Mathews. I also wish to acknowledge Dr. Rex Thomas, Dr. Ann Thompson, and Dr. Tom Andre. These three inspired me to think deeply about the ideas underlying this thesis topic.

My wife Amy also deserves credit for this work. I was motivated by her encouragement and her personal standards of excellence.

Finally, this thesis is dedicated to my eight year old niece Amy Joe, and her younger brothers BJ and Adam. These children and their classmates will be among the first to experience the learning environments supported by microcomputer technology. It is my hope these experiences will be exciting and active ones.

APPENDIX A.

HUMAN SUBJECTS FORM

INFORMATION ON THE USE HUMAN SUBJECTS IN RESEARCH
IOWA STATE UNIVERSITY
(Please follow the accompanyin 77 structions for completing this form.)

① Title of project (please type): Effects of integrating activities on transfer of skills and retention of knowledge.

② I agree to provide the proper surveillance of this project to insure that the rights and welfare of the human subjects are properly protected. Additions to or changes in procedures affecting the subjects after the project has been approved will be submitted to the committee for review.

Greg Davis — Typed Named of Principal Investigator
2/2/84 — Date
Signature redacted for privacy — Signature of Principal Investigator

N31 Quadrangle — Campus Address
4-6840 — Campus Telephone

③ Signatures of others (if any) — Date — Relationship to Principal Investigator

Signature redacted for privacy — 2-3-84 — Signature redacted for privacy

Signature redacted for privacy — 2-3-85 — Signature redacted for privacy

④ ATTACH an additional page(s) (A) describing your proposed research and (B) the subjects to be used, (C) indicating any risks or discomforts to the subjects, and (D) covering any topics checked below. CHECK all boxes applicable.

☐ Medical clearance necessary before subjects can participate
☐ Samples (blood, tissue, etc.) from subjects
☐ Administration of substances (foods, drugs, etc.) to subjects
☐ Physical exercise or conditioning for subjects
☐ Deception of subjects
☐ Subjects under 14 years of age and(or) ☐ Subjects 14-17 years of age
☐ Subjects in institutions
☐ Research must be approved by another institution or agency

⑤ ATTACH an example of the material to be used to obtain informed consent and CHECK which type will be used.

☐ Signed informed consent will be obtained.
☒ Modified informed consent will be obtained. (See Attached pre-test)

⑥ Anticipated date on which subjects will be first contacted: Month Feb Day 20 Year 1984

Anticipated date for last contact with subjects: May ___ 1984

⑦ If Applicable: Anticipated date on which audio or visual tapes will be erased and(or) identifiers will be removed from completed survey instruments:

Month ___ Day ___ Year ___

⑧ Signature of Head or Chairperson — Date — Department or Administrative Unit

Signature redacted for privacy — 2/7/84 — Signature redacted for privacy

⑨ Decision of the University Committee on the Use of Human Subjects in Research:

☒ Project Approved ☐ Project not approved ☐ No action required

George G. Karas — 2/9/84 — Signature redacted for privacy

APPENDIX B.

PRECOURSE ASSESSMENT

Secondary Education 101
Preliminary Assessment
Spring 1984
Ann D. Thompson

DIRECTIONS: This instrument is designed to assess your current experience with and knowledge of computers. The results will be utilized in planning experiences for this class and for baseline data as we continue to assess computer knowledge of incoming students in future classes.

Results on this instrument will not affect your grade in Sec. Ed. 101.

PART I: (Background)

NAME ________________________________

MAJOR ________________________________

Year (Please circle) Fr. Soph. Jun. Sen. Grad.

College ______________________________

Sex (Please circle) M F

Age __________

PART II: (Previous Experiences)

1. Briefly describe length and content of previous computer courses (or parts of courses) you have taken in high school or college.

	Course or Unit Name	Length of Course or Unit	School/Group Offering Course
(1)	________________	____________________	______________________
(2)	________________	____________________	______________________

(1) Major Topics Included in Course or Unit ______________________________

____

(2) Major Topics Included in Course or Unit ______________________________

____

2. Briefly describe any other experiences you have had working with computers. For example, have you visited a computer store, do you have a computer at home, have you used a computer at work?

3. Describe the major reasons you are enrolled in Sec. Ed. 101X.

PART III: (Attitude)

Using the answer sheet supplied, please record your reactions to the following items. (a) = low (c) = medium (e) = high

Part A — Lo — Medium — Hi

1. GENERAL INTEREST. Compared to other students you associate with at the University, how do you rate your own interest in computers?

|______|______|______|______|

2. PRESENT ABILITY. Compared to other students (not necessarily computer science majors)... the "average" or "typical" students, how do you rate your own present knowledge and ability when it comes to computers?

|______|______|______|______|

3. COMPUTERS AS A HOBBY. Compared to photography, stamp collecting, sailing, playing cards, weaving, or other hobbies, how do computer games, programming, and other computer activities stack up?

|______|______|______|______|

(a) (b) (c) (d) (e)

Lo Medium Hi

4. COMPUTERS AS APPLIANCES. Compared to dishwashers, telephones, TV, pocket calculators, or other things, "...we may not be able to get along without..." where do computers place in the scheme of things?

5. COMPUTERS AND GENERAL EDUCATION. How important or valuable do you feel computers are as a part of general education for a college student like yourself?

6. COMPUTERS IN BUSINESS. How important are computers for the person in business such as the salesperson, farmer, or operator of a small retail store?

7. COMPUTER ANXIETY. How would you rank your anxiety, fear, or general feeling of helplessness when it comes to dealing with computers?

8. COMPUTER "TALENT"
How much natural ability or talent do you feel you have (in comparison to others around you) in working with computers, programming, and general computer operation?

9. COMPUTERS AND JOB-SEEKING. How valuable do you feel computer literacy will be in giving you an edge over others who may not have computer literacy when it comes to getting a job after you graduate?

10. COMPUTERS AND SOCIETY. What role do you perceive computer will be playing in our culture in the next few years, in terms of impact and influence?

PART IV The following items are designed to provide a brief assessment of your current knowledge of computers. For each item, select the one most appropriate answer and record your answer on the answer sheet provided. There is no penalty for guessing.

11. A CRT

a. is used to display input
b. is used to display output
c. resembles a typewriter keyboard
d. both a and b

12. Another word for I/O device is

a. peripheral
b. terminal
c. CPU
d. chip

13. A binary digit is called a

a. bite
b. bit
c. byte

CHOOSE THE MOST CORRECT DEFINITION OF THE FOLLOWING KEY WORDS. (14&15)

14. Computer software

a. a chip
b. micro-processor
c. computer programs
d. flexible materials

15. Analog

a. compares objects
b. continuously measures physical conditions
c. opposite structure
d. similar structure

16. A digital computer

a. accepts and counts specific units
b. accepts continuous units
c. both a and b

17. A new development called firmware

a. eliminates the need for external input devices
b. is programmed into the computer
c. will probably replace software
d. is available only for APPLE computers

18. A string variable is usually

a. numbers and blanks only
b. numbers, letters and blanks
c. letters and blanks only

19. A string variable is identified through the use of

a. #
b. $
c. *

20. The command REM is used by a programmer to

    a. document the program
    b. add comments
    c. identify variables

21. The basic command NEW:

    a. Clears all data from memory
    b. Assigns values and variables
    c. Sets values and their order
    d. Comments not to be executed
    e. Removes line(s) from memory

22. FORTRAN is a computer programming language designed primarily for

    a. recreational applications
    b. scientific applications
    c. educational applications
    d. analog applications
    e. business applications

23. COBOL is a computer programming language designed primarily for

    a. recreational applications
    b. scientific applications
    c. educational applications
    d. analog applications
    e. business applications

24. PASCAL is a computer programming language designed primarily for

    a. recreational applications
    b. scientific applications
    c. educational applications
    d. analog applications
    e. business applications

25. Machine language is

    a. a low level computer language
    b. a high level computer language
    c. based on base 10 numeration
    d. the first language learned by most programmers

26. The computer-related job closest to that of a typist is:

    a. computer operator
    b. keypunch operator
    c. systems analyst
    d. computer programmer
    e. I don't know

27. In order to program a computer, a person:

- a. can use any English language words
- b. can use any English or foreign language words
- c. must use programming language numbers, not words
- d. must use the words from a programming language
- e. I don't know

28. Choose the correct output for the computer program shown below:

```
10  LET C = 6
20  LET D = 8
30  LET E = C+D+2
40  PRINT E
50  END
```

Output

- a. 6
- b. 14
- c. 8
- d. 16
- e. I don't know

29. When were computers first manufactured in large numbers?

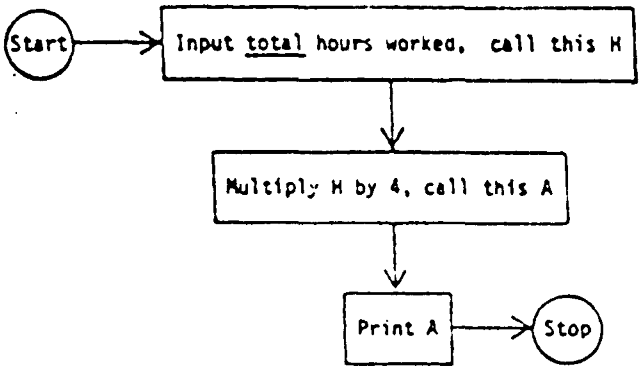
- a. 1860's
- b. 1890's
- c. 1920's
- d. 1950's
- e. I don't know

30. What is the main purpose of the following program:

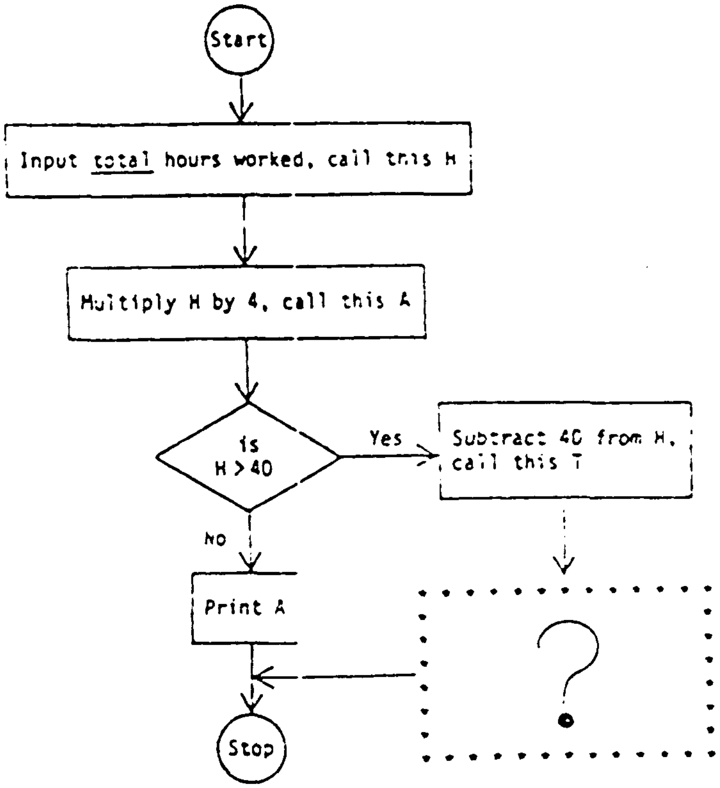
```
10  INPUT A, B, C, D, E
20  LET S = A+B+C+D+E
30  LET M = S/5
40  PRINT S,M
50  END
```

- a. store A, B, C, D, and E in the computer
- b. print the letters S and M
- c. print the sum and average of five numbers
- d. calculate large sums
- e. I don't know

31. A flowchart to determine the weekly wages of employees in a bakery is shown below. Employees are paid $4 per hour up to 40 hours per week.

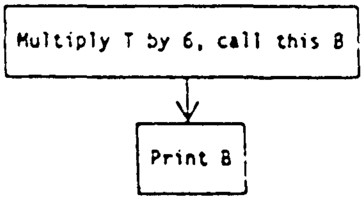


Employees are now to be paid "time-and-a-half" ($6 per hour) for overtime (hours worked over 40). How would you complete the flowchart below to include overtime pay? Select answer a, b, c, d, or e.

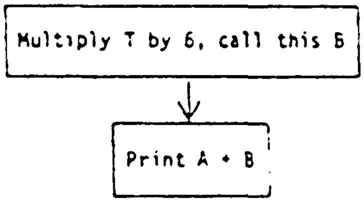


ANSWERS

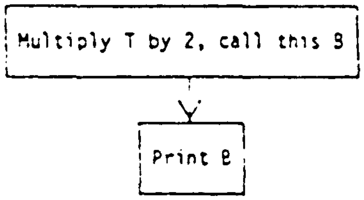
a.



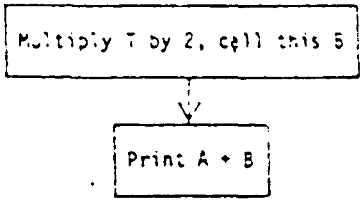
b.



c.



d.



e. I don't know

APPENDIX C.

PRETEST

NAME________________

LOGO PRE-TEST

1) In lab this week, you will be using LOGO's turtle graphics capabilities to create regular polygons. This semester we are experimenting with two different forms of instruction to teach this material. The results of this study will influence future 101 instructional design. Both forms of instruction used in this study have been used successfully in 101 before.

The study consists of several activities and tests, which you will all complete. Your performance on these activities will be the basis of the study's results. Performance on these activities will not directly affect your 101 course grade.

THIS INFORMATION WILL BE KEPT STRICTLY CONFIDENTIAL. Please indicate your consent to include your performance in our data by selecting (a) below.

a) I give my consent.

b) I do NOT give my consent.

2) Which of the following commands produce the same result as RT 90?

- a) LT 90
- b) LT 270
- c) RT 360
- d) RT 540

3) The command SAVE "SQUARE would

- a) save a procedure titled square on the disk.
- b) save a procedure titled square in a disk file titled square.
- c) save all procedures in memory in a disk file titled square.
- d) produces a syntax error.
- e) none of the above.

4) If you want to edit a procedure titled square, what do you type in?

- a) TO SQUARE
- b) SQUARE
- c) FIX SQUARE
- d) ERASE SQUARE

5) The command READ "SQUARE would

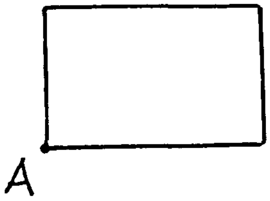
- a) take the procedure SQUARE out of the RAM and place it in active memory.
- b) read the procedure SQUARE from the disk.
- c) read the procedures stored in the disk file titled SQUARE from the disk.
- d) produces a syntax error.
- e) none of the above.

6) When working on a LOGO project, you should

- a) break the project into small, workable steps.
- b) program th entire project, then break it into smaller components.
- c) work only with simple geometric designs.
- d) all of the above.

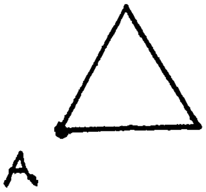
7 – 10) For each of the figures below, determine the number of degrees you would turn if you traveled from point A along the figure's perimeter back to point A, ending in the position you started.

7)



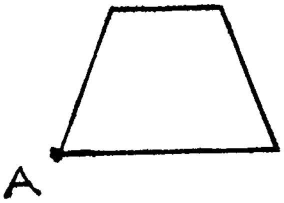
a) 180 b) 720 c) 360 d) not a,b or c e) don't know

8)



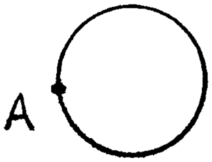
a) 180 b) 720 c) 360 d) not a,b or c e) don't know

9)



a) 180 b) 720 c) 360 d) not a,b or c e) don't know

10)



a) 180 b) 720 c) 360 d) not a,b or c e) don't know

APPENDIX D.

POSTTEST PART 1

NAME____________________

SECTION__________________

LOGO **POST TEST**

PART ONE

1) To the best of your ability, restate the Total Turtle Trip Theorem.

2) Which of the following commands produce the same result as RT 90?

a) LT 90
b) LT 270
c) RT 360
d) RT 540

3) The command SAVE "SQUARE would

a) save a procedure titled square on the disk.
b) save a procedure titled square in a disk file titled square.
c) save all procedures in memory in a disk file titled square.
d) produces a syntax error.
e) none of the above.

4) If you want to edit a procedure titled square, what do you type in?

a) TO SQUARE
b) SQUARE
c) FIX SQUARE
d) ERASE SQUARE

5) The command READ "SQUARE would

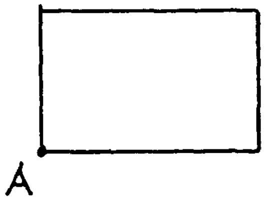
a) take the procedure SQUARE out of the RAM and place it in active memory.
b) read the procedure SQUARE from the disk.
c) read the procedures stored in the disk file titled SQUARE from the disk.
d) produces a syntax error.
e) none of the above.

6) When working on a LOGO project, you should

a) break the project into small, workable steps.
b) program th entire project, then break it into smaller components.
c) work only with simple geometric designs.
d) all of the above.

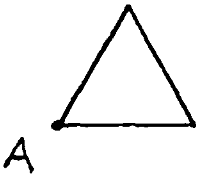
7 - 10) For each of the figures below, determine the number of degrees you would turn if you traveled from point A along the figure's perimeter back to point A, ending in the position you started.

7)



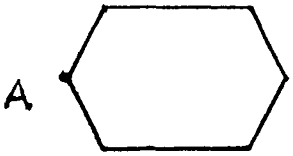
a) 180   b) 720   c) 360   d) not a,b or c   e) don't know

8)



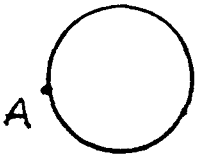
a) 180 b) 720 c) 360 d) not a,b or c e) don't know

9)



a) 180 b) 720 c) 360 d) not a,b or c e) don't know

10)



a) 180 b) 720 c) 360 d) not a,b or c e) don't know

APPENDIX E.

POSTTEST PART 2

NAME________________

SECTION_____________

START TIME: ______________________

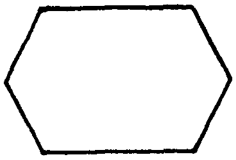
LOGO POST TEST

PART TWO

1) Draw a figure where the sum of the turns is 360°

2) Write a procedure that will draw a ten sided figure.

3) Write a procedure that will draw the figure below.



4) Here are the procedures you worked on last week.

```
TO HEXAGON
REPEAT 6 [FD 40 RT 60]
END

TO TRIANGLE
REPEAT 3 [FD 40 RT 120]
END

TO PENTAGON
REPEAT 5 [FD 40 RT 360/5 ]
END

TO CIRCLE
REPEAT 360 [FD 1 RT 1 ]
END
```

Since that time, you have seen how to use variables with LOGO. Re-write each of the above procedures, including a variable (:L) for the length of a side of the polygon.

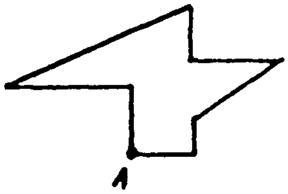
5) Write a procedure that will draw any regular polygon, with any length side. (HINT: You will need to use two variables, one for side length, and one for the number of sides of the polygon you want to draw.) PLEASE SHOW ALL YOUR WORK.

END TIME:____________ TOTAL TIME (MINUTES):______________

APPENDIX F.

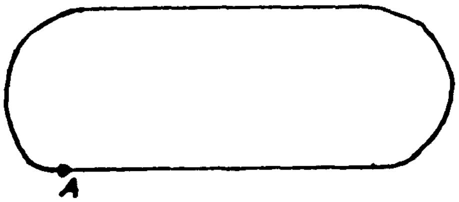
RETENTION QUESTIONS

58. In the following figure, if you start at point A and trace the figure, ending at point A in the position you started in, how many degrees have you turned? (Assume a left turn is counted as a negative value.)



a) 360 - (sum of the left turns)
b) 360 + (sum of the left turns)
c) 360
d) None of the above

59. In the following figure, if you start at point A and trace the figure, ending at point A in the position you started in, how many degrees have you turned?



a) 180
b) 720
c) 380
d) None of these

APPENDIX G.

CONTROL GROUP TREATMENT

The Total Turtle Trip Theorem

In this activity you will be examining Logo procedures based on The Total Turtle Trip Theorem. Here is the theorem's definition:

> If a turtle takes a trip around the boundary of any area and ends up in the state in which it started, then the sum of all turns will be 360°.

Next week in lab, you will be tested over the theorem, as well as the procedures you work with in the following activities.

Name____________
Soc. #__________

TOTAL TURTLE TRIP EXERCISES

In each of the following five activities you will be given a LOGO procedure that draws a polygon. You will be asked to enter the procedure, and then edit that procedure to produce varied results. Determine if the Total Turtle Trip Theorem holds for each case.

STARTING TIME:_________

Activity 1

You've already seen this first procedure in your lab manual....

```
TO SQUARE
REPEAT 4 [FD 20 LT 90]
END
```

In this procedure, does the turtle end up in the state it started? (Yes , No) Is the sum of all turns 360 . (Yes , No)

Enter the procedure, but write it so it draws a square with a side of 50. Does the Total Turtle Trip Theorem still hold? (Yes , No)

Activity 2

Enter the following procedure :

```
TO HEXAGON
REPEAT 6 [FD 30 RT 60]
END
```

If the FD 30 is changed to a FD 55, would the Total Turtle Trip Theorem hold? Why or why not?

Change the FD 30 to FD 55 and see if the theorem actually holds true.

Activity 3

In the following theorem does the Total Turtle Trip Theorem hold true. (Yes , No)

```
TO TRIANGLE
REPEAT 3 [FD 30 RT 60]
END
```

Enter the procedure and see if it draws a polygon.... How would you change the procedure to draw a polygon? Remember, the total turtle trip theorem must hold true....

Now that you have a good triangle procedure, change to FD 30 to FD 5. Does the Total Turtle Trip Theorem still hold?

Activity 4

Here is a procedure that draws a circle:

```
TO CIRCLE
REPEAT 360 [FD 1 RT 1]
END
```

Enter this procedure and run it. Does the turtle end in the same state which it started? (Yes , No) How many degrees did the turtle turn? ______

At this time try several different values in the FD part of the circle procedure.

Activity 5

In this procedure which draws a pentagon, the number for the RT command has been omitted....

```
TO PENTAGON
REPEAT 5 [FD 40 RT ___ ]
END
```

What should the missing number be? _____
List the steps you took to arive at your answer.

Enter the procedure (if you haven't already) and test your answer.

STOP TIME:________ TOTAL TIME(MINUTES):________

APPENDIX H.

EXPERIMENTAL GROUP TREATMENT

The Total Turtle Trip Theorem

In this activity you will be examining LOGO procedures based on **The Total Turtle Trip Theorem.** Here is the theorem's definition:

> If a Turtle takes a trip around the boundary of any area and ends up in the state in which it started, then the sum of all turns will be 360°.

Next week in lab, you will be tested over the theorem, as well as the procedures you work with in the following activities.

NAME________________
SOC. #______________

TOTAL TURTLE TRIP EXERCISES

START TIME:_______________

Recall the total turtle trip theorem which states if the turtle ends in the state it began, it has turned 360 . For example, in the SQUARE procedure (listed in the lab manual), the turtle ends in the state it began.

```
TO SQUARE
REPEAT 4 [FD 20 RT 90]
END
```

Looking closely, we can see the turtle completes four 90 degree turns (for a total of 360°). Thus the total turtle trip theorem holds true.

In the following activities you will be asked to create four procedures. For each of these procedures the total turtle trip theorem will hold true. When you feel you have met this requirement for a particular procedure, HAVE THE LAB INSTRUCTOR CHECK IT BEFORE PROGRESSING TO THE NEXT PROCEDURE!

Activity 1- Write a procedure that draws a hexagon (six sides).

Activity 2- Write a procedure that draws a triangle (three sides).

Activity 3- Write a procedure that draws a circle.

Activity 4- Write a procedure that draws a pentagon.

In each of the above procedures, does the total turtle trip theorem hold? (Yes , No)

END TIME:_______________ TOTAL TIME (MINUTES):________

APPENDIX I.

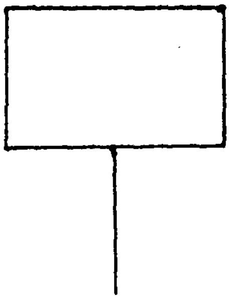
LOGO HOMEWORK ASSIGNMENT

LOGO LAB #1 HOMEWORK

For the next lab, write procedures that draw the figures below. Save all the procedures you write in one disk file titled FIGURES.

Problem 1:

Problem 2:



Problem 3: