The effect of learning problem-solving methods on learning to program in the BASIC language

Yen-chu Hung
Iowa State University
INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI
A Bell & Howell Information Company
300 North Zeeb Road. Ann Arbor, MI 48106-1346 USA
313/761-4700  800/521-0500
The effect of learning problem-solving methods on learning to program in the BASIC language

by

Yen-chu Hung

A Dissertation Submitted to the
Graduate Faculty in Partial Fulfillment of the
Requirements for the Degree of
DOCTOR OF PHILOSOPHY

Department: Industrial Education and Technology
Major: Industrial Education and Technology

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

Iowa State University
Ames, Iowa

1995
The effect of learning problem-solving methods on learning to program in the BASIC language

Yen-chu Hung

Major Professor: William G. Miller
Iowa State University

This study was designed to compare learning problem-solving methods versus non problem-solving activity (word-processing) on subsequent learning to program in the BASIC language. It also examined a method to provide students with increased knowledge and skills to enable them to learn how to program.

A pretest-posttest control group design was used in this experiment with random assignment of subjects to one of three groups. Experiment groups one (deduction group) and two (induction group) first received the pretest and learning problem-solving methods; then group one received deduction instruction while group two received induction instruction, both followed by learning BASIC language programming instruction, taking midterm test one and two, and then the post-test. The control group first received the pretest and wordprocessing instruction, followed by learning BASIC language programming instruction and taking midterm test one and two, and then the post-test.

The results indicated that when female students first study problem-solving methods (induction and deduction) they experience a significant increase in BASIC
language programming achievement. Likewise, male students who first learn problem solving (induction) experience a significant increase in BASIC language program achievement.

The study also showed that female students who first receive problem-solving instruction in induction subsequently learn BASIC language programming significantly better than female students who first receive problem-solving instruction in deduction and subsequently learn BASIC language programming.

Further evidence supports that female students in group one and two on BASIC language programming in design and understanding performed significantly better than female students in the control group. In addition, male students who first learn problem solving (induction) perform significantly better than males who first receive non-problem solving instruction prior to learning BASIC language programming in design and understanding.

From this study, the researcher concluded the following: (1) students who first learn problem-solving methods, rather than receiving non problem-solving instruction followed by learning BASIC programming, perform significantly better a their counterparts; and (2) female students who learn problem solving (induction) perform significantly higher than female students who learn problem solving (deduction) followed by learning BASIC language programming. Thus, first learning problem-solving skills enhances the ability to learn a programming language.
# TABLE OF CONTENTS

## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table of Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>v</td>
</tr>
</tbody>
</table>

## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Table of Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>vii</td>
</tr>
</tbody>
</table>

## CHAPTER I. INTRODUCTION

1. **Background of the Study** ............................................. 1
2. **Basic Theoretical Understandings** ................................ 3
3. **Purpose of the Study** .............................................. 6
4. **Objectives of the Study** .......................................... 6
5. **Question of the Study** ............................................. 7
6. **Assumptions of the Study** ......................................... 8
7. **Limitations of the Study** ......................................... 8
8. **Procedures of the Study** ......................................... 9
9. **Definition of Terms** .............................................. 10

## CHAPTER II. REVIEW OF LITERATURE

1. **Learning Theory** ...................................................... 12
   1. **Learning** .......................................................... 12
   2. **Behavioral reinforcement** ..................................... 14
   3. **Cognitive processing and memory** .......................... 15
   4. **Transfer of learning** ........................................... 16
2. **Learning to Program** ................................................. 18
   1. **BASIC language programming** ................................ 18
   2. **Why teach programming** ....................................... 19
   3. **Computer language features** ................................ 21
   4. **Programming design skill** .................................... 23
3. **Problem Solving** ..................................................... 25
   1. **Definition of problem solving** ............................ 25
   2. **Problem-solving method - deduction** ..................... 29
   3. **Problem-solving method - induction** ...................... 31
   4. **Metacognition in problem solving** .......................... 32
   5. **Gender difference and problem solving** .................. 33
4. **Learning problem-solving Methods by Heuristics** ............ 33
   1. **Heuristics** ...................................................... 33
   2. **Heuristics in problem solving** ............................. 34
   3. **Heuristics and induction** .................................... 35
5. **Does Learning Problem Solving Subsequently Affect the Ability to Learn Computer Programming** ............................ 36
   1. **The influence of prior programming experience on problem solving** ............................ 36
The relationship between BASIC language programming and problem solving .............................................................. 37
Effect of problem solving methods on learning to program .......................................................... 38
Does learning induction and deduction affect ability to learn programming ......................................................... 40
Learning model .................................................................................................................................................. 41

CHAPTER III. METHODOLOGY .................................................................................. 45
Overview of the Experimental Research Design .................................................................................. 45
Population and Sample .................................................................................................................. 47
Group distribution of .................................................................................................................. 47
Measurement .................................................................................................................................................. 49
Data collection instruments ........................................................................................................ 49
Validation of the pretest and post-test .................................................................................................. 51
Reliability of the pretest and post-test ................................................................................................. 52
Assessment instruments .................................................................................................................. 53
Variables of the Study .......................................................................................................................... 54
Independent variables .................................................................................................................. 54
Dependent variables .................................................................................................................. 54
Covariate Variables .................................................................................................................................. 54
Experimental Treatments .................................................................................................................. 55
Laboratory equipment .................................................................................................................. 55
Instructional materials .................................................................................................................. 55
Experimental group .................................................................................................................................. 56
Hypotheses of the Study .................................................................................................................. 57
Data Collection and Analysis ........................................................................................................ 67
Statistical treatment .................................................................................................................. 67
Analysis .................................................................................................................................................. 69

CHAPTER IV. RESULTS .......................................................................................... 70
Descriptive statistics .................................................................................................................. 70
Inferential statistics .................................................................................................................. 74
Pretest hypotheses .................................................................................................................. 74
Midterm hypotheses .................................................................................................................. 75
Post-test hypotheses .................................................................................................................. 79
Hypotheses regarding BASIC post-test in design and understanding .............................................. 82
Correlation hypotheses .................................................................................................................. 85
Findings .................................................................................................................................................. 86
Summary .................................................................................................................................................. 90
CHAPTER V. SUMMARY, DISCUSSION, CONCLUSIONS, AND RECOMMENDATIONS ........................................... 91
   Summary of the study ........................................... 91
   Discussion ..................................................... 93
   Conclusions ................................................... 96
   Recommendations .............................................. 98
      Recommendations for teachers college .................... 98
      Recommendations for future research ..................... 100

REFERENCES ....................................................... 101

ACKNOWLEDGEMENTS ............................................ 110

APPENDIX A. HUMAN SUBJECTS REVIEW ............................ 112

APPENDIX B. PRETEST MATERIAL .................................. 116

APPENDIX C. POST-TEST MATERIAL ................................. 125

APPENDIX D. INSTRUCTION MATERIAL ............................ 135

APPENDIX E. COVER LETTER AND SURVEY INSTRUMENT ....... 144

APPENDIX F. BASIC LANGUAGE PROGRAMMING MODULES ...... 146
**LIST OF TABLES**

Table 4.1: Mean and standard deviations for the BASIC language pretest .. 70
Table 4.2: Mean and standard deviations for the problem-solving pretest .. 71
Table 4.3: Mean and standard deviations for the midterm one test on BASIC language programming .......................... 71
Table 4.4: Mean and standard deviations for the midterm two test on BASIC language programming .......................... 72
Table 4.5: Mean and standard deviations for the BASIC language post-test .......................... 72
Table 4.6: Mean and standard deviations for the problem-solving post-test .......................... 73
Table 4.7: Mean and standard deviations for the BASIC language post-test in program design .......................... 73
Table 4.8: Mean and standard deviations for the BASIC language post-test in program understanding .......................... 74
Table 4.9: Summary of one-way ANOVA for pretest means scores .......................... 75
Table 4.10: Summary of one-way ANOVA on the midterm examinations .......................... 76
Table 4.11: Results of midterm test one scores .......................... 77
Table 4.12: Results of midterm test two scores .......................... 78
Table 4.13: Results of midterm test two scores for female group .......................... 78
Table 4.14: Results of midterm test two scores for male group .......................... 78
Table 4.15: Summary of ANCOVAs for post-test examinations .......................... 79
Table 4.16: Problem-solving post-test .......................... 80
Table 4.17: Problem-solving post-test scores for the female group .......................... 80
Table 4.18: BASIC language post-test scores ........................................ 81
Table 4.19: BASIC language post-test scores for the female group ............ 81
Table 4.20: BASIC language post-test scores for the male group .................. 81
Table 4.21: Summary of the ANCOVAs for BASIC post-test in design and understanding ................................................................. 82
Table 4.22: BASIC language post-test scores in design ............................ 83
Table 4.23: BASIC language post-test scores in design for the female group ................................................................. 83
Table 4.24: BASIC language post-test scores in design for the male group ................................................................. 83
Table 4.25: BASIC language post-test scores in understanding .................. 84
Table 4.26: BASIC language post-test scores in understanding for the female group ................................................................. 84
Table 4.27: BASIC language post-test scores in design for the male group ................................................................. 85
Table 4.28: Summary of correlations between the BASIC language pretest and post-test; problem-solving pretest and post-test .... 85
LIST OF FIGURES

Figure 1.1: Subset of cognitive elements found in problem solving and programming ........................................... 4
Figure 2.1: Learning mapping .................................................. 13
Figure 2.2: Behavioral reinforcement map .................................. 15
Figure 2.3: Architecture of human information processing system .... 17
Figure 2.4: A mapping for the analysis of learning BASIC program .... 42
Figure 3.1: Instruction schedule ............................................... 46
Figure 3.2: Group distribution ................................................... 48
Figure 4.1: Means for the problem-solving pretest and post-test ....... 87
Figure 4.2: Means for the BASIC language pretest and post-test ....... 88
Figure 4.3: Means for the BASIC language pretest and post-test (male) .... 88
Figure 4.4: Means for the BASIC language pretest and post-test (female) ... 89
CHAPTER I. INTRODUCTION

Background of the Study

Computers are very useful in educational systems today. Many schools have purchased or are purchasing microcomputers for teaching (IBM-PC, Macintosh, etc.). At most school levels, students have been instructed in various programming languages. When using higher-order thinking skills, these students are able to develop powerful ideas about thinking through the process of programming. There may be a high relationship between programming language instruction and higher-order thinking skills such as problem solving. For example, programming skills could improve learners' mathematics study skills and help them in designing computer programs. Through metacognition, learners are able to correct small problems in their procedures (e.g., "debugging" or "stepwise refinement") while finding a solution to a programming problem.

Widespread individual studies have been conducted in schools on computer programming using the LOGO language and its possible impact on cognitive development. Salomon and Perkins (1985) listed six areas in which transfer of learning from programming might occur: (a) mathematical and geometric concepts and principles; (b) problem solving, problem finding, and problem management strategies; (c) abilities in formal reasoning and representation; (d) models of knowledge, thinking, and learning; (e) cognitive styles; and (f) enthusiasm and
tolerance for meaningful academic engagement. Au and Leung (1991) suggested that LOGO training has beneficial effects on children's higher level cognitive skills such as problem solving. Dalton and Goodrum (1991) suggested that, when used together, computer programming and problem-solving strategy instruction may provide an effective means of teaching transferable problem-solving skills. Papert (1980) said that learning computer programming with LOGO is an ideal environment for learning problem-solving skills and increase the learner cognitive activity. These studies have shown that LOGO computer programming is an ideal environment for learning problem-solving skills because the LOGO language has (a) a top-down programming design; (b) modularity; and (c) it requires limited use of logical constructs.

The findings of studies about other computer languages also confirm that programming portrays an ideal environment for learning problem-solving skills. Funkhouser and Dennis (1992) indicated the effects of problem solving computer software on increasing problem-solving ability. Reed et al. (1987; 1988) found that learners using both the LOGO computer language and the BASIC computer language had significant increases in problem-solving skills. They also found no significant difference between using the LOGO language and the BASIC language in increasing problem-solving skills.

Several studies have been conducted regarding the implication of cognitive psychology related to designing programs. Hooper (1986) reported that students
using computer programming simulation employed more sophisticated algorithms during programming than did students who were not exposed to the manipulative computer model (MEMOPS) which was designed to facilitate the learning of programming. Thomas and Hooper (1991) reported that simulation may be useful for reinforcing complex sequences. When using simulations the learner is forced to assume responsibility for executing the process, whereas in the alternative methods the learner responds to external questions or instructions. Alperson and O’Neil (1990) compared a computer-based tutorial with simulations for transmitting knowledge in beginning anthropology and psychology courses. Salisbury (1990) found that in the cognitive psychology area, the development of the use of subskills, inference, spaced practice, spaced review, the capacity of short-term memory, and the representation of information in memory are related to each issue in the design of computer drill programs.

Basic Theoretical Understandings

While the literature reviewed in Chapter II strongly suggests that learning a computer programming language may improve a learner’s problem-solving abilities, it is quite possible that the converse may also be true; that is, developing problem-solving skills may enhance the ability to learn a programming language. Both “problem solving” and “programming” involve a common subset of cognitive behaviors, memorizing and a schema or template. It may be said that both provide
a set of experiences which enhance the learning of the other. One might graphically depict this situation as shown in Figure 1.1. If a learner is deficient in the cognitive structure common to both programming and problem solving proficiency, he or she must acquire these structures. It was suspected by the researcher that, for success in such activities, the learner must acquire these shared structures.

Figure 1.1. Subset of cognitive elements found in problem solving and programming
The mode for instruction in either activity may simply involve spending sufficient time and reinforcing the activities that build common cognitive structures.

Of course, the identification, description and validation of the common elements or structure are, by themselves, a major research problem. Support for this theory of "mutual causation" may be given by demonstrating that instruction in either problem solving or programming will enhance the acquisition of skill in the other.

Some studies have been reported on learning problem-solving methods which could increase student learning to program in the BASIC language. Palumbo and Reed (1992) found a significant, positive correlation between problem-solving skills and BASIC language competency measures. Bayman and Mayer (1988) reported learning BASIC programming involves the growth of syntactic and conceptual knowledge and that strategic knowledge and problem solving performance are strongly related to measures of conceptual knowledge. Palumbo and Reed (1991) showed that some problem-solving skills of high school students can be increased through systematic exposure and interaction with the BASIC programming language. The current study examined the effect of learning problem solving prior to learning to program in the BASIC language.
Purpose of the Study

The purpose of this study was to compare the effects of learning problem-solving methods with instruction in a non problem-solving activity on subsequent achievement in learning to program in the BASIC language. This research may suggest that learning problem-solving methods could be important for students in providing them with increased knowledge and skills to learn how to program.

Objectives of the Study

The objectives of this study were to:

1. Evaluate the effectiveness of problem-solving instruction given to students in a teacher's college who were also learning BASIC language programming.
2. Compare the BASIC language achievement levels of college students who received problem-solving instruction first versus students who received non problem-solving instruction.
3. Identify which of two problem-solving instructional methods best helped students to better understand the design of programs in the BASIC language.
4. Compare the achievement of male and female students receiving different problem-solving methods in different groups.
5. Relate problem-solving instruction to learning BASIC language programming.
Questions of the Study

This study sought to answer the following questions:

1. Will there be significant differences among experimental and control groups on the BASIC language program pretest mean scores?

2. Will there be a significant difference between the problem-solving pretest mean scores of the experimental and control groups?

3. Will there be a significant difference between the adjusted post-test means of the experimental and control groups on BASIC language achievement?

4. Will there be a significant difference between the adjusted post-test means of the experimental and control groups on problem-solving achievement?

5. Will there be a significant difference between the adjusted post-test means of the experimental and control groups on BASIC language programming design?

6. Will there be a significant difference between the adjusted post-test means of the experimental and control groups on BASIC language program understanding?

7. Will there be a significant relationship between the BASIC language tests and the problem-solving tests?
Assumptions of the Study

This study was based upon the following assumptions:

1. Student achievement scores were normally and independently distributed in the two experimental and control groups with respect to ability in problem-solving learning and wordprocessing learning.

2. The effect of the teachers was approximately the same in the experimental and control groups.

3. No interaction (social, academic, or otherwise) occurred among students outside of the experimental setting which may have affected the results of the study.

4. A random assignment of subjects by group to either experimental or control groups minimized the effects of extraneous variables.

Limitations of the Study

Several limitations were imposed on this study:

1. It was limited to the Microsoft Quick BASIC language and the students who were enrolled in the computer BASIC concepts class held during the spring semester of the 1995 school year at National Chayi-I Teachers College in Taiwan.

2. Because the course in learning to program is a full semester course in Taiwan, one full semester was needed to collect data.
3. The experiment was conducted in one school and, therefore, may not be
generalizable to subjects in other schools.

4. Only two problem-solving methods were used for the experimental groups.

5. Wordprocessing material was used by the control group.

6. Seventy-five students participated in this study.

7. This study was limited to selected laboratory and classroom activities.

8. This study was limited to instruction provided by three instructors.

9. The materials covered in this study were related to BASIC language
   programming with practical applications.

10. The textbook used in this study was *Fundamentals of Quick Basic language
    programming* by Hung, gin-quar and published by the Su-kung Computer Book

**Procedures of the Study**

The procedures of the study consisted of the following:

1. Formulated the study problem.

2. Reviewed related literature concerning problem-solving and BASIC language
   programming.

3. Identified the population and sample for the study.

4. Developed pretest and post-test instruments.

5. Administered the pilot test.
6. Analyzed the pilot test data using the SPSS and SAS computer software packages.

7. Developed and refined the pretest and post-test instruments.

8. Administered the pretest.

9. Implemented instruction.

10. Administered the post-test.

11. Coded the research data.

12. Analyzed the post-experiment data using the SAS package.

13. Interpreted the findings.

14. Wrote the summary, conclusion, and recommendations.

Definition of Terms

*BASIC language* - One of the simplest computer languages. The word BASIC is an acronym for Beginners All-purpose Symbolic Instruction Code. This language, written in 1964 at Dartmouth College by Kemeny and Kurtz, allows humans to give instructions to a computer.

*Computer-assisted instruction (CAI)* - The use of a computer as an aid in a classroom setting to enhance student learning.

*Concept* - A specific set of objects, symbols, or events that share common characteristics and can be referred to by a specific word or symbol.
Deduction problems - Premises are given and the problem solver must apply the appropriate rules to draw a conclusion.

Heuristics - Strategies that assist students, understanding and advance their use of resources in solving problems.

Induction problems - A series of instances are given and the problem solver must induce a rule or pattern that describes the structure of problem.

Problem solving - Cognition science suggests three characteristics for the definitions of problem solving (Mayer, 1983):

1) Problem solving is cognitive, but is inferred from behavior.
2) Problem-solving results in behavior that leads to a solution.
3) Problem solving is a process that involves manipulation of or operations on previous knowledge.

Programming test - A test that a student attempts to implement to solve a program.

Transfer - Transfer of learning occurs “Whenever prior-learned knowledge and skills affect the way in which new knowledge and skills are learned and performed” (Cormier & Hagman, 1987, p. 1).
CHAPTER II. REVIEW OF LITERATURE

The purpose of this review is to explore the relationship between problem-solving knowledge and understanding BASIC language programming. This review is organized into the following five sections: (1) Learning Theory; (2) Program Learning; (3) Problem Solving; (4) Learning Problem-solving Methods by Heuristics; and (5) The Subsequent Effect of learning Problem Solving on learning Computer Programming.

Learning Theory

Learning

Learning is the process in which a human changes his or her behavior as the result of experience obtained from a teacher or from a learning environment—external or internal. Bransford (1979) and Mayer (1981) defined meaningful learning as integrated learning, a "... process in which the learner connects new material with knowledge that already exists in memory" (p. 121). Michalski (1989) said the learning process includes the learner's background knowledge, and is motivated by the learner's desire to reach a goal such as to solve a problem, to understand a fact or observation, or to finish a task. Michalski's learning model is shown as Figure 2.1. Further, a change in behavior occurs in the process of learning. Behavior refers to some action (i.e., muscular or mental) or combination of
actions. Obvious behaviors such as talking, writing, moving, and the like allow one to study cognitive behaviors—thinking, feeling, wanting, remembering, problem solving, creativity, etc. Typically, in an academic setting, the change in behavior one may be looking for is the ability to remember, understand and apply various concepts, and the tendency to exhibit certain attitudes and values of the kind set forth in educational objectives.

The concluding fundamental variables in the definition of learning are experience and an interchange with the environment whereby stimuli take on meaning and relationships are established between stimuli and responses. Thus, one can say that:

\[
\text{Learning} = \text{experience} + \text{inferencing}
\]

(Modified by Goforth, 1994, p. 1)
Behavioral reinforcement

Reinforcement plays a key part in learning, because behavior that is continued by reinforcement is strengthened (Skinner, 1968). A reinforcer is any event that can be displayed to strengthen a response. Thorndike (1898) studied the behavior of cats placed in a box whose door could be opened if a lever was pressed. After the experiment, the cat would come to be able to escape immediately upon entering the box. Thorndike explained the cat's problem solving by invoking the concepts that have come to be called reinforcement and extinction. Thorndike called this learning process trial-and-error learning (Figure 2.2) and used this basic model as a description of problem solving in general. Thorndike's view has provided the basic behavioral model of problem solving (Campbell, 1960; Davis, 1973; Mayer, 1983). Trial-and-error learning occurs when a stimulus situation demands a response, but the correct response is not dominant in the response for that situation. Travers (1965) noted that, during programmed learning, the student is questioned at each stage and is rewarded immediately with the pleasure of being correct. Such immediate reinforcement aids in the retention of knowledge.
Cognitive processing and memory

Atkinson and Shiffrin (1968) first proposed that there are complicated mental processes between a stimulus and a response. People use their information-processing memory to help them think about how they think, or how they process information cognitively. This model differentiates between the external world and internal cognitive processing.

Stimuli or information from the environment is stored in one's short-term sensory storage (STSS). Anderson (1980) noted that this is an especially important point in the process because what one recently is processing as working memory is a fundamental factor in determining the stimuli one encounters. If one does not pay attention to new information coming in, it is forgotten; whereas, Phye & Andre (1986) suggested if one pays attention to it, the stimuli moves from short-term sensory storage (STSS) to short-term memory (STM) and a working memory (WM).
storage system. Short-term memory is conscious memory—all that one is conscious of at one time. Information in STM or WM, if rehearsed or encoded, remains the focus of attention or is passed along to long-term memory (LTM). The information stored as long-term memory is almost never forgotten, although one may be unable to regain it because of a failure in a way to search for it (Figure 2-3.). Furthermore, knowledge located in long-term memory is important in problem solving. According to Hayes (1981), "If you are missing relevant knowledge, an easy problem may appear difficult or impossible . . . Much that passes for cleverness or innate quickness of mind actually depends on specialized knowledge" (iv-v). While one's short-term memory may handle about seven "chunks", the capacity of long-term memory is thought to be virtually limitless (Frederiksen, 1984). This has been demonstrated through the applications of problem-solving theory to the writing process.

**Transfer of learning**

The process that enables one to use previously learned responses in new situations is called transfer. Cormier and Hagman (1987) defined transfer of learning as that which occurs "... whenever prior-learned knowledge and skills affect the way in which new knowledge and skills are learned and performed" (p. 1).
A conceptual model consists of words or diagrams used in instruction to help learners build mental models of what they are studying. What conceptual models do well is to highlight the main ideas or objects or actions in a system. They are accurate and useful representations of knowledge that are needed when solving problems in some particular domain.

Students who build a personal mental representation based upon the conceptual model presented to them, and who learn to manipulate these models
cognitively, appear to be far better able to solve transfer tasks. They rely on the models of learning as a background for analyses. Pirolli (1986) conducted research among high school students and found that students are able to transfer knowledge obtained from instructional examples of programs to new skills for programming recursive functions, and to transfer these cognitive skills across programming problems.

Learning to Program

BASIC language programming

There are two main reasons why the researcher chose to study classes in BASIC language instead of other programming languages. First, the BASIC language has been used widely in classrooms. While more sophisticated computer languages such as C++, Pascal, Fortran, COBOL, etc., exist, they are neither easily accessible nor obtainable throughout all school levels, especially high school or elementary school. Because BASIC is a language widely available in both the IBM compatible and Macintosh markets, it exists in nearly every computer and programs can be transferred as ASCII text files. This ease of implementation has resulted in a high percentage of utilization. BASIC has also been used since the earliest developmental stages of personal computers (e.g., in the 8080 computer) to the Pentium level of system design today. It is a clear benchmark of this technology.
Second, research has shown that learning to program the computer in BASIC could improve problem-solving ability (Rose, 1984). A study conducted by Putnam et al. (1984) evaluated a population of 96 high school students who received a test of BASIC programming concepts and were individually interviewed after a semester course in BASIC programming. While the high school students had numerous misconceptions prior to understanding BASIC programming, their ability to achieve computer skills in programming and problem solving were greatly enhanced following instruction in BASIC programming.

Why teach programming?

Three rationales are offered for the teaching of programming: (1) to provide students with a start on fundamental skills to expand their reasoning ability and problem-solving skills that will be as important as reading and arithmetic in their later lives and occupations (Galanter, 1984); (2) to promote computer literacy (Luehrmann 1984); and (3) to become more viable participants in the workplace with the diversity to provide solutions to everyday problems (Scheffler, 1986). The third rationale provides students with an understanding of continuous improvement concepts being used today in industry. In addition, programming has the potential to expand the intellectual capabilities of learners, making them inventors of their own intellectual tools, amplifying the powers of humanity with a capacity to liberate human potential.
Programming gives a computer a set of systematic instructions so that it can perform a certain task. This task might be drawing a picture, giving a quiz, writing a poem, playing a game, etc. A common misconception is that doing computer programming is somehow similar to doing arithmetic. This is not accurate. Mathematics or the science of mathematics is based on laws and theorems which have been widely accepted, while the technology of the computer continues to expand the avenues of technology. The student-learning activities that are most like programming are essentially building blocks and making up stories, or in today's terminology, they are the cybernetic application of man's capabilities. Like block builders, programmers combine a small number of elements in a fairly structured way to reach a main goal.

Haigh (1985) suggested that most instructional software are already doing a lot to assist students to develop the skills needed to do programming and learn to use a computer effectively. If one recognizes the potential of these software systems and can also see interrelationships, then one can see the potential for using BASIC to expand teaching. Instructors could lead students to see new connections and relationships. The potential is said to be limitless. While this power prevails across all software programs, the mere fact that the BASIC computer language is available on every computer makes it most commanding across nationalities and global boundaries.
Computer language features

A novice programming course could present far more information to a student than just learning a language. It should teach the student about the discipline and nature of a well-structured program. Mayer (1981) reported that much learning which results from computer programming is knowledge gained regarding the features and functions of the language. This knowledge correlates with recognizing and interpreting the validity of statements in a computer language, and knowing how to build correct statements (Pea et al., 1987). Instruction in language features gives definitions and examples of the right statements. To learn programming concepts, one must understand the mechanics of programming functions to more clearly understand correct language statements and software features, and optimize the reformation of a quality programming language.

Gannon (1976) reported that understanding and reformulation of computer language features are based on activities which are the benchmark of programming courses. In theory, the student can learn programming from a book, change the output messages in programs, modify arithmetic expressions, change the loop number, and make other changes to the design to make a workable program. Brooks (1980) said the most qualified programmer will learn how to enhance and improve the efficiency of program applications by removing program activities and loops that cause the software to seek and hunt, and inhibit the throughput time for the operation. Hence, the knowledge of language features is not a general use
requirement but is critical for the development and application of sound program
design and structure for the optimal utilization of software.

Hung et al. (1992) reported that the growing demand for the application of
computers has affected the introduction of programming courses in teachers
colleges in Taiwan. No longer can programming be only an introduction to the
general concept, but it must focus on the critical design tools which allow the
programmer to write an efficient program. Aligned specifically in the college
curriculum, computer programming is a relatively new phenomenon, and there may
be a limited understanding of how students learn to program. Difficulties and
misconceptions in programming may play an important role in the development of
the learner's understanding. Learning through making "mistakes" may ultimately
lead to a deeper level of skill in programming.

The mental models that novice programmers develop as they learn to
program in BASIC have been studied by Bayman and Mayer (1983). It has been
suggested that students need to be introduced to actual computer architecture,
logical data placement, and signature location where information is stored—e.g.,
high memory, extended memory, etc. Understanding, addressing, showing, and
recognizing mapping of data in the computer make it possible for information to flow
through the computer more efficiently. Shneiderman (1976) reported the novice
student needs to be able to follow, in a step-by-step fashion, how data (information)
flows under the control of the language commands in order to complete a software
application. Also, students need to be encouraged to role play what the computer is doing, how it responds to their commands, and interpret each command task in pursuance of the program. This agrees with research conducted about conceptual bugs in novice programming (Pea, 1986). Pea's research found that the best way to help students is to provide clear models that show how the processing of control and data is arranged by a specific programming language. Given such a model, students tend to write better programs and learn from positive reinforcement.

**Programming design skill**

Skinner's method of operant conditioning provides the learner with an appropriate model that could be used for programmed instruction. It is a system whereby each student is presented with content and is required to respond actively and in a positive manner where immediate feedback is given about the correctness of the response. In the classroom, programmed instruction has led to renewed efforts to achieve the ideals of individualization and to treat students according to their unique abilities, individual learning rate, and interests. Dalbey and Linn (1985) reported that learners using programmed instruction have more opportunity to work at their own speed, to make positive strides, and to gain first-hand appreciation for their ability to communicate in a language foreign to them but necessary to be effective in a computerized environment.
Many researchers (Lehrer & Randle, 1987; Miller & Emihovich, 1986; Pea & Kurland, 1984) agree that the design of a program is a cognitive activity. Programming design skills are techniques used to design and refine new programs. The student uses a mental model to learn the program. Simple computer models allow the learner to mentally track the movement of data through a program. By tracking the data; the learner builds concepts of computer's operations which are used in program development. Such concepts are displayed in models and procedural programming skills recognized in a global community and used by professional programmers. Adaptation of these concepts by students enhances their ability to design their own computer programs.

The construction of computer models is achieved when students learn about designing a program. Bayman and Mayer (1983) reported that learners may begin to understand typical algorithms such as loops in writing simple programs, thus allowing them to create similar programs to serve their basic needs. With practice and application, students tend to improve their skill as programmers. One would say these stereotypic patterns used in designing software serve as models which reflect flexible and powerful techniques to build their own specialized application programs. Such applications of models tend to reduce the cognitive demands of programming because they decompose complicated procedures into easy (mechanical) structures. Pea (1986) remarked that models enable students to
design programs in a top-down (i.e., logical flow) manner and let them write procedures that can be chained together to reflect more complicated tasks.

In this step-by-step process students develop procedural skills that, when achieved, give them command of a programming language. The student/learner will become proficient and capable to design, write, and apply the software in its entirety. To go one step beyond, the most accomplished student will plan commands that combine and optimize the software to get desired functions. The best students will test the resulting program and challenge themselves to find errors. Beyond the programming level of writing language statements, the most capable students will take on the task of debugging to cancel errors that impede or inhibit the quality of the program. However, students need to gain experience using a programming language before they acquire procedural skills to a level which will allow them to be recognized as programmers.

**Problem Solving**

**Definition of problem solving**

Problem solving is the means by which an individual uses previously acquired knowledge, skills and understanding to satisfy the demands of an unfamiliar situation (Krulik & Rudnick, 1980). Problem solving is based on common knowledge and everything one brings to the table at the time in which a decision must be made. This, coupled with the power of teams to make decisions, greatly
improves the possibility of seeking solutions or resolutions with the expedience needed in a global market. While the impact may sometimes be greatly exaggerated, the solutions are very basic.

Problem solving emerges as a process involving a set of skills which can and should be taught. According to Krulik and Rudnick (1984), it is accomplished by employing a series of steps:

1. Read the problem
2. Explore
3. Select a strategy
4. Solve the problem
5. Review, look back, and extend the solution

Problem solving consists of the mental and behavioral activities that are involved in dealing with problems. Problem solving may involve thinking (cognitive) components, emotional (or motivational components), and behavioral components (Andre, 1994).

One simple model of problem solving identifies five critical cognitive processes: identification, definition, exploration, action, and looking and learning—an ideal approach to problem solving (Berliner & Gage, 1992). Davis (1973) defined a problem solution as a creative idea or a new combination of existing ideas.
A broad array of literature, ranging from experimental psychology to social and organizational psychology, has been concerned with efforts to describe and prescribe the process of solving problems. Dewey (1933) described two steps in problem solving. First, problem solving is a state of doubt or difficulty, and second, it becomes an act of searching and inquiring to find information to resolve doubt or difficulty (Sinnott, 1989).

Problem solving refers to the whole process from problem detection through various attempts to problem solution or problem abandonment. It refers to the overall process of responding to a problem (Gilhooly, 1989):

1. Detecting that a problem exists, realizing there is a discrepancy between the current situation and a goal, and that a solution cannot be reached without a further investigative search.

2. Formulating the problem more completely, to have a more clear understanding of the problem.

3. Using a more detailed approach enables constructing a representation of the variables and providing optimal approaches to solutions (i.e., stating the problem, reducing or limiting the scope, and dealing more directly with individual variables) so attempts for resolution can be identified and made.

Two basic schools of thought in problem solving are behaviorist and information processing. The behaviorist approach deals with the stimulus (input) response (output) aspects of problem solving (Skinner, 1968). The information-
processing approach deals with the process that intervenes between input and output, and leads to a desired goal from an initial state (Rubinstein, 1975).

Generally, it seems the human mind, when confronted with a problem, goes through seven key stages in reaching a solution (Laurire, 1990):

1. Absorb and understand the statement.
2. Make some immediate inferences, so far as possible.
3. "Play with" the situation.
4. Reflect and let things mature.
5. Look for a better representation and frame a closed statement.
6. Find a partial solution and return to item 2, or complete the solution.
7. Check the validity of the solution—generalize.

The most useful definitions to problem solving from cognitive science are among those offered by Bartlett (1932), Davis (1984), and Mayer (1983). The work of all three researchers postulates problem solving as a multiple-step process in which the problem solver must find relationships between past experience (schema) and the problem at hand.

Mayer (1983) was especially helpful in developing an instrumental definition of problem solving and suggested three characteristics for definitions of problem solving:

1. Problem solving is cognitive, but is inferred from behavior.
2. Problem-solving results in behavior that leads to a solution.
3. Problem solving is a process that involves manipulation of or operations on previous knowledge.

Problem solving is the process of figuring which set of past experiences (schema) best relates to the problem at hand. The problem solver must interpret the new situation based on the schema selected and then act upon that match to find a solution. One of the most viable tools today to enhance this process is nested in the computer and the power of the software available. As stated earlier, the BASIC language, having a global identity, appears to be a useful tool to implement the solving of problems at the speed of a computer.

**Problem-solving method - deduction**

Discrimination is the ability to determine the objects and/or events that have a direct impact on the problem. Students can go beyond the simple practice of discrimination and be led to solve classification problems and develop logical thinking skills in interesting ways. Skinner (1968) noted that deduction is a way of constructing discriminative stimuli. Useful forms of deduction inspire the thinker to formulate a systematic form of analysis which will reduce the problems that exist to the simplest form. For example, a deductive inference could be stated as: A personal computer has a 100 MHz processor. One can deduce that this computer is faster than computers with 33 MHz processors since 33 is less than 100.
Programming often involves arranging statements into a logical sequence; for example,

\[
\begin{align*}
\text{Input } X, Y \\
Z &= X + Y \\
\text{Print } Z
\end{align*}
\]

is logical because before one can add and print the sum, the values must first be entered. The sequence

\[
\begin{align*}
Z &= X + Y \\
\text{Input } X, Y \\
\text{Print } Z
\end{align*}
\]

would not be logical. Deductive reasoning would lead to this conclusion because of prior experiences in attempting to present a result without having the necessary information. By arranging a problem into a series of logical steps, one applies "deductive" reasoning. Therefore, computer language programming requires extensive application of the principles of deduction. This computer programming strategy can be developed in several ways. Most appropriate to this research is to expand the students' ability to use deduction, which subsequently affects one's ability to design and understand computer language programming.
Problem-solving method - induction

Induction requires increased emphasis when considering its potential for creativity in solving new problems. Inductive thinking should expand one's considerations and remove barriers of fixed-rule thinking. Developing a knowledge base will improve the chance of finding a solution, and having the ability to draw on a broader knowledge base. According to Luchins and Luchins (1970) induction, in the inductive reasoning process, considers many new facts that are related to what is already known and a new whole as is formed. Creative solutions are demanded of inductive thinkers. Such thinkers will have to organize, retrieve, and use an excess of information to solve their problems. Induction uses experimental reasoning to arrive at the whole from the particulars. The initial formulation of constructive induction uses domain knowledge to develop a new concept or attributes, beyond those supplied in the input (Michalski, 1983). Thus, induction employs basic inference strategy used in synthesized learning.

Creativity should not be construed to be limited to only inductive thinking (Wertheimer, 1945). Creative efforts may involve both deductive and inductive thinking in the solution of problems of expression. Using problem-solving methods as a skill to understand and design computer language programming is the highest order commonly found in the demonstration of a designer's idea and expression of his or her aesthetic feelings. Thus, problem solving involves mostly intuitive, creative thought. A broad understanding of problem-solving capabilities and the
logical combination of induction as a skill along with one's personal knowledge broadens one's abilities to design and understand programming.

**Metacognition in problem solving**

Much domain-specific knowledge is related to the ability to solve problems. Metacognitive ability suggests another effective means of influencing problem-solving ability. Metacognition is the awareness of thinking processes (i.e., thinking about one's thinking). Therefore, thinking how one thinks about finding a solution is as important in the design of seeking answers as the variables that must be analyzed. The metacognitive approach uses reflective thinking, or cognitive monitoring. Schoenfeld (1985) noted that performance on many tasks is positively correlated with the degree of one's metacognitive ability. If students are taught to monitor their cognitive thoughts which they use to try to solve new programs, they learn to focus and understand their own thinking strategies. Understanding one's own thinking becomes a major tool in problem solving according to Simon (1980). Students are required to be aware of how these strategies are built within memory to provide a repertoire of problem-solving actions. Metacognitive strategies can be used as tools to help students learn problem-solving skills. These results suggest that teaching students metacognitive skills could help them to develop problem-solving skills.
Gender differences and problem solving

Research by Benbow and Stanley (1980), and Ethington and Wolfe, (1986) reported that gender was found to have a moderate causal effect on problem-solving achievement. These studies also found that males typically outscore females in mathematics and mathematics problem solving. Gallagher and DeLisi (1994) reported that females were more likely to use conventional problem-solving strategies, and the use of conventional strategies was associated with negative mathematics attitudes.

Learning Problem-solving Methods by Heuristics

Heuristics

Heuristics are strategies that assist students' understanding and advance their use of resources in solving problems. Heuristics are often applied to future events that will ensue and where there exists no acceptable data base applicable to an unmet need which may be encountered. These problems exist in social (e.g., discrimination), moral (e.g., pornography), and physical (e.g., combating viruses) environments. Polya (1957) promoted the widespread use of heuristics and believed that problem-solving ability is a series of principles that are systematically applied to a relative database of knowledge. Polya suggested four broad strategies to help a learner acquire problem-solving skills: (1) understand the problem; (2) devise a plan; (3) carry out the plan; and (4) look back.
Polya's heuristics initiated research for strategies designed to help students learn to be successful problem solvers. Polya (1957) believed that problem solving can be learned, and it could happen in many domains. Polya's ideas were examined by other researchers as described below.

Schoenfeld (1985) declared that the heuristic strategies suggested by Polya were only labels for categories of related strategies, and did not lead to specific procedures in seeking solutions. Schoenfeld claimed that considerable procedural knowledge was needed and heuristics appeared to generate useful strategies for solving problem.

Krulik and Rudnick (1980) developed their own more specific mathematical heuristics with greater clarification. Krulik and Rudnick's mathematical heuristics were used in the present study to assist students to increase their mathematical problem-solving skills.

**Heuristics in problem solving**

Leightton (1989) reported using heuristics in a learning situation in which seventh grade students were separated into four groups. The first group (control) received no treatment. The other three groups were divided as follows: (1) those working individually; (2) those working in groups; and (3) those working as cooperative teams. The three treatment groups received instruction related to heuristic strategies, and afterward they were given a problem-solving post-test.
Results of the post-test showed there were no significant differences between the three treatment groups; however, there was a significant difference in the problem-solving capabilities of the control group.

To use heuristics to learn problem solving, students must not only understand heuristic strategies but also how to apply them. These students are usually identified as being self-directed and self-motivated, and have the ability to design critical thinking and problem-solving scenarios.

Krulik and Rudnick (1980), reported that students use heuristic strategies to develop problem-solving skills. As a result, teachers need to spend time showing students how and when to use heuristics. Simon (1980) showed that students need to become conscious of how heuristic strategies are built within memory and applied when necessary. Weigand (1991) also reported heuristics fits within current teaching practices. Therefore, more attention needs to be given to problem solving to decide how heuristics and experimental strategies are presented to learners.

**Heuristics and induction**

Heuristics invite many diverse strategies. According to Polya (1957), heuristics employ analogy, induction, figures, generalization, deduction, etc. Heuristic reasoning is often based on induction or analogy. Induction is the process of discovering general law by the observation and combination of particular
instances. Induction is used in all science, and it is easily manipulated and quantified in a computerized environment.

Does Learning Problem Solving Subsequently Affect the Ability to Learn Computer Programming?

The influence of prior programming experience on problem solving

Adelson (1981; 1984) indicated that experts and novices represent problems and solve them differently. Adelson noted that prior programming experience reflects a significant difference when addressing problems and seeking solutions. Students with no prior programming experience have greater difficulty in problem solving. Thus, prior knowledge of programming can influence problem solving.

Several studies concur that prior knowledge of computer languages improves problem-solving ability (Choi & Repman, 1993; Jones, 1988; McCoy & Orey, 1988). It could be said that learning computer programming increases problem-solving skill. Shneiderman and Mayer (1979) reported that computer programming tasks such as generating, modifying, and debugging computer programs are problem-solving activities. Thus, it may be inferred that an increase in problem-solving skills could also transfer to design and understanding computer language programming. This is the premise of the present research study.
The relationship between BASIC language programming and problem solving

If one understands the infusion of the BASIC programming language in the computer world, it is hard to ignore the BASIC language as a common communication tool among professionals. Several researchers have supported and acknowledged the impact of the BASIC language as a tool for problem solving. Their research related to BASIC programming activities supports the premise that learning this language contributes to developing and increasing problem-solving skills and techniques with resulting cognitive benefits.

Mayer et al. (1986) found that four supposedly programming-specific cognitive skills were more highly predictive of performance in primary BASIC than three non specific skills. In addition, they found that training in programming specific procedural skills prior to a programming course enhanced the ability to learn BASIC language. Conversely, is learning computer programming an effect subsequent to learning the problem-solving method? It is hypothesized in this study that learning problem-solving skills leads to a significant improvement in cognitive skills and has a direct positive relationship with learning the BASIC computer programming language.

Palumbo and Reed (1991; 1992) found that, in a structured BASIC programming environment, students did not show a significant increase in four measures of problem-solving ability (i.e.,) after 45 hours of programming
experience. After 90 hours of programming experience, there were significant increases in the students’ problem-solving scores.

Research on the effectiveness of structured programming languages (e.g., Logo and Pascal) on increasing problem-solving skills has shown no significant outcomes over BASIC language programming. Reed et al. (1987; 1988), used the components of the Ross Test of Higher Cognitive Processes and the Watson-Glaser Critical Thinking Appraisal to test problem-solving skill ability of students and found significant increases in both the Logo and BASIC programming group. They also found no significant difference between the gains in either group. These findings substantiate the use of the BASIC language in the current research.

**Effect of problem-solving methods on learning to program**

To teach programming to students, research has shown a need for a more structured form of instruction to express the concept of program design. Linn (1985) and Mayer (1988) suggested that planning specific programs will effectively enhance learning computer language design. Plans can build program fragments that symbolize model action sequences in programming with particular tasks or subtasks. Researchers have used an expert model of programming to teach a beginning structured language such as PASCAL (Linn & Clancy, 1989; Soloway, 1986). Expert models of programming rely on plans as a central idea.
It is not adequate to have merely a theoretical plan for programming. One must develop a reinforcing method for delivery that supports utilization of heuristics to improve inductive, deductive, metacognitive, and creative thinking as methods to apply when attempting problem solving. Programming schemata effectively requires the student to understand all the tools available and to use them in the most expeditious way to seek resolution. Vosniadou & Ortony (1989) found that students who are exposed to analogs with surface similarity and deep similarity could induce a schema by the process of mapping. Then, the induced schema (i.e., mapping identities) form the basis for analytical reasoning and problem solving. Based on this premise, problem solving can be taught and learned effectively. Problem solving helps students comprehend computer programming problems deeply which, in turn, helps them solve problems efficiently. Greeno and Simon (1988) reported that patterns of information in a problem have to be recognized to determine that a problem-solving operation can be applied. McKeithen et al. (1981) found that expert problem solvers represent problems immediately in terms of core programming structures which allow them to find elegant solutions. On the other hand, novice problem solvers fixate on surface features of problems without comprehending the structure which can be translated into a program. The current study deduces that programming skills are an effective approach to teach problem-solving skills to improve students' ability to understand and solve critical thinking problems faced daily. In other words, students' problem representation and
problem solving in teaching programming would be significantly improved by teaching schemata of programming.

Salomon & Globerson (1987) reported that the degree of students' mindfulness as a tendency influences their learning. The more mindful a student is, the greater the capacity to learn computer programming. It can be inferred that if students can be subsequently taught to use computer programming design after learning problem-solving methods, their ability to learn programming design will develop more rapidly. Such an understanding led this researcher to validate this study. Could it be that designing and understanding BASIC programming can be learned more rapidly by those who first use heuristics to learn problem-solving skills?

Does learning induction and deduction affect ability to learn programming?

Kahney (1993) noted that induction and deduction are opposite problem-solving methods. Induction involves some set of cognitive processes that enables one to abstract rules from experience. Learning proceeds from the specific to a general rule. On the contrary, deduction is the set of processes used to apply rules that one has previously acquired knowledge. Learning is acquired by using general knowledge to solve a specific problem. Thus, induction is associated with the learning process, whereas deduction is associated with the application of knowledge. Pea (1986) found that the best way to help students to learn computer
programming is to provide clear models that show the process of controlling data. One could infer that students learn deduction (problem-solving skill) by using a step-by-step method to solve problems. Then, after learning computer programming, such as employing top-down model programming, one could use deductive methods or ideas to design and understand computer programming. Papert (1980) and Feurzig et al. (1981) reported that metacognition, general problem solving, and divergent thinking (inductive reasoning) have possible cognitive benefits toward active participation in computer programming by students.

Snow (1980), Ricardo (1983), and Langstaff (1989) noted that inductive reasoning ability influences programming achievement. Since students show differences in cognitive abilities (inductive reasoning ability) and mindfulness in computer learning, problem solving affects new task cognitive skills. These skills are directly related to design of programs. Thus, if one teaches students problem-solving methods using induction or deduction, one could also increase their ability to understand and design computer language program. This was hypothesized in the present study.

The learning model

An approach commonly used and addressed by researchers to teach programming is the learning model. For each programming lesson, topics are read in textbook materials and then problems are solved with the guidance of the BASIC
language concept. It is assumed that students use their prior knowledge such as problem solving, induction, and deduction to explain and interpret the available instructional texts and examples. These interpretative and explanatory methods employed in teaching the BASIC language produce declarative knowledge that may become stored in the learners' memories. Kintsch (1986) and van Dijk and Kintsch (1983) reported students' processing of instructional texts and examples increased through the use of recent textbook processing models. In addition, Johnson-Laird (1983) reported models employed in the learners' objectives build a congruous mental model which is used to interpret the given material. This knowledge gained is due to the interaction of the presented material and the construction of interpretation strategies retained by students.

![Diagram](image)

**Figure 2.4.** A mapping for the analysis of learning BASIC program (Modified by Piroli & Recker, 1994, p. 239)
In the learning model, the students are given sets of programming problems as exercises using the BASIC language after reading instructional materials (see Figure 2.4).

To complete the exercise problems included in most texts requires some combination of familiar and new subtasks. Some familiar subtask situations need to access formerly acquired cognitive skills. Singley & Anderson (1989) reported that practice improves the effectiveness of applying cognitive skills. Other programming problem exercises may use new subtask situations. Newell (1990) reported these situations cause problem solving impasses that can be resolved through the use of problem-solving methods such as analogy, induction, and interpreting declarative knowledge obtained from reading the instructional text and examples. Thus, declarative knowledge is used to search for new answers in problem solving. The effectiveness of problem solving in new situations is conditional on the declarative interpretations built by individuals for the given instructional texts and examples.

Problem solving at these impasses affect the grasp of new task cognitive skills represented as production rules. The subsequent new structure and avenues of problem-solving result from impasse resolution. New production knowledge is directly related to problem solving at impasses. New declarative knowledge (a resolution) about the programming design domain is subsequently improved by achieving this resolution. Therefore, learners reflect on their problem solving or the
structure for their solutions, and expand new boundaries in the area of program learning.
CHAPTER III. METHODOLOGY

This chapter describes the experimental research design which used quantitative measures to examine the effect of learning problem-solving methods on learning computer programming. A three-group, controlled experiment following the static-group comparison design (Campbell & Stanley, 1963) was conducted. First, the overview of the experimental research design is described. Then, the population and sample, sample assignment to group, measurements, variables of the study, experimental treatments, and hypotheses of the study are described. Finally, the data collection and analyses are discussed.

Overview of the Experimental Research Design

This study involved three groups of students enrolled in a basic computer concept course at National Chayi-i Teachers College in Taiwan. The fundamentals of the BASIC language program (i.e., loops, input/output, etc.) were covered in this class, and the resulting achievement of the three groups were compared. The instruction schedule for the three group over 16 weeks is shown in Figure 3.1. The first group received problem-solving instruction first, then deduction instruction, followed by BASIC language program instruction. The second group received problem-solving instruction first, then induction instruction, followed by BASIC language program instruction.
<table>
<thead>
<tr>
<th>Group</th>
<th>1 Deduction</th>
<th>2 Induction</th>
<th>3 Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BASIC programming &amp; problem-solving pretest</td>
<td>Problem-solving instruction</td>
<td>Wordprocessing</td>
</tr>
<tr>
<td>2-3</td>
<td>Problem-solving instruction</td>
<td>Problem-solving instruction</td>
<td>Wordprocessing</td>
</tr>
<tr>
<td>4</td>
<td>Deductive instruction</td>
<td>Inductive instruction</td>
<td>Wordprocessing</td>
</tr>
<tr>
<td>5-7</td>
<td>BASIC programming instruction</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>BASIC programming instruction</td>
<td>BASIC programming midterm test one</td>
<td></td>
</tr>
<tr>
<td>9-11</td>
<td>BASIC programming instruction</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>BASIC programming instruction</td>
<td>BASIC programming midterm test two</td>
<td></td>
</tr>
<tr>
<td>13-15</td>
<td>BASIC programming instruction</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>BASIC programming instruction</td>
<td>BASIC programming &amp; problem-solving post-test</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3.1. Instruction schedule
The third group received wordprocessing instruction first, followed by BASIC language programming instruction.

Population and Sample

The population of this study was college freshman students from Chayi-i Teachers College in Taiwan. The sample consisted of three intact freshman classes in the school randomly assigned to the three treatments. The 93 students in the three classes who participated in the study were enrolled in courses offered through the Elementary Education Department during the Spring of 1995. Of the 93 students initially enrolled, 75 participated in this study. The data from the remaining 18 students were not included because these students had previously studied a computer language and this experience may have influenced them to develop greater problem solving and programming ability (Choi & Repman, 1993; Jones, 1988; McCoy & Orey, 1988).

Group Distribution

The distribution of male and female students were n = 31 and n = 62. The first class was selected to be control group; the total number was 30, or male (n = 15) and female (n = 15). Of the 30 students initially enrolled, 24 participated in this study. The data from the remaining 6 students, male (n = 5) and female (n = 1) were not included because these students had previously studied a computer language. The second class was selected to be the deduction group. The total number of 33
included male (n = 1) and female (n = 32). Of the 33 students initially enrolled, 32 participated in this study. The data from the remaining 1 male student was not included because this student had previously studied a computer language. The third class was selected to be the induction group. The total number of 30 included male (n = 15) and female (n = 15). Of the 30 students initially enrolled, 19 joined in this study. The data from the remaining 11 students, male (n = 6) and female (n = 5), were not included because these students had previously studied a computer language (see Figure 3.2.). A total of three groups were taught by three teachers. To avoid bias due to the teacher factor, the researcher made arrangements for each teacher to instruct the three groups concurrently using the same teaching material.

Each teacher taught the BASIC language course for four weeks.

<table>
<thead>
<tr>
<th>Group Number</th>
<th>1 Deduction</th>
<th>2 Induction</th>
<th>3 Control</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Male</td>
<td>Female</td>
<td>Male</td>
</tr>
<tr>
<td>Initially enrolled</td>
<td>15</td>
<td>15</td>
<td>1</td>
</tr>
<tr>
<td>total</td>
<td>30</td>
<td>33</td>
<td>30</td>
</tr>
<tr>
<td>Eliminated</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>total</td>
<td>6</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Participated</td>
<td>10</td>
<td>14</td>
<td>0</td>
</tr>
<tr>
<td>total</td>
<td>24</td>
<td>32</td>
<td>19</td>
</tr>
</tbody>
</table>

Figure 3.2. Group distribution
Measurements

Data collection instruments

Because of the need to measure the level of achievement in the experimental and control groups, several tests (pretests, mid-terms, and post-tests) were given to the subjects during class. Prior to administration, copies of the tests were submitted for approval by the Human Subjects Review Committee at Iowa State University to ensure that the rights and welfare of the human subjects were adequately protected. Then an informed consent was obtained from National Chiayi Teachers College following the appropriate procedures. Both signed approval forms are shown in Appendix A.

Pilot test - Pilot testing included 43 Chayi-i Teachers College junior level students to test the four initial test versions (i.e., the problem-solving pretest, problem-solving post-test, design BASIC programming pretest, and BASIC programming post-test). The tests were held at Taiwan Chayi-i Teachers College before the experiment was conducted, and the results were sent back for analysis (i.e., reliability, and the correlation between the pretest and post-test on problem solving and BASIC language programming). The instruments were then modified as necessary.

Pretest - This instrument was developed by the researcher and given within the first week of the experiment. The problem-solving pretest consisted of 20
questions designed to measure students' problem-solving ability. The BASIC
language program pretest consisted of 15 programming concept questions in
understanding BASIC commands and designing programs. The pretest questions
were related to the textbook materials (see Appendix B for the English version).

Post-test - This instrument was also developed by the researcher and given
during the last week of the experiment. The problem-solving post-test consisted of
20 questions designed to measure the students' problem-solving ability. The BASIC
language programming post-test consisted of 20 program questions in
understanding BASIC commands and designing programs. The post-test questions
were related to the textbook materials. A copy of the post-test appears in Appendix
C.

Exams - Two exams were given to the subjects, the first during week 8 and
the second during week 12. These two midterm tests contained 1 and 7 questions
related to BASIC language programming, respectively. One hour was designated
for completion of each exam. The midterm test one score used 1 (right answer) or 0
(wrong answer). The reliability of midterm test one was not calculated because it
consisted of only one question. The midterm test two score was 100 points (first
three questions each 10 points, question 4-5 each 15 points, and question 6-7 each
20 points). The reliability of midterm test two as calculated by the Kuder-
Richardson formula 21 was 0.961. (Note: k = 100 and the total mean and standard
deviation were used to get the number.)
Validation of the pretest and post-test

According to the American Psychological Association (1985): “Validity ... refers to the appropriateness, meaningfulness, and usefulness of the specific inferences made from test scores. Test validation is the process of accumulating evidence to support such inferences” (p. 58). Isaac & Michael (1990) noted content validity is especially important for achievement and proficiency measures, and for measures of adjustment or social behavior based on observation in selected situations.

The following steps were used to measure the validity of the test in this study. First, the four tests (English version) were reviewed by graduate students in the Departments of Computer Science, Curriculum and Instructional Technology, and Industrial Education and Technology at Iowa State University. Their comments were incorporated to improve the content validity of these four tests. Then, before the pilot test, the four tests (Chinese version) were reviewed by a group of 20 knowledgeable persons, including three computer language programming teachers, two psychology teachers, and 15 elementary education graduate students of the National Chiayi Teachers College in Taiwan. Their suggestions were included in the final revision of the experiment test.
Reliability of the pretest and post-test

Reliability is often measured using Cronbach’s coefficient alpha presented in the Statistical Package for the Social Sciences (SPSS). A reliability analysis of the four tests was undertaken in this study: the BASIC language programming pretest and post-test; and the problem-solving pretest and post-test. The internal consistency reliability score was 0.64 for the BASIC language programming pretest. The internal consistency reliability was 0.86 for the BASIC language programming post-test. The correlation of the BASIC language pretest and post-test was 0.84, which was significant at the 95% confidence level. The reliability score was 0.53 for the problem-solving pretest, and 0.51 for the post-test. The correlation of the problem-solving pretest and post-test was 0.57, which was also significant at the 95% confidence level. No pilot test data were obtained for the exam tests which were developed by instructors who were responsible for the three sections of subjects in the study. The correlation between the pretest and post-test were slightly higher than one would expect given the reliability obtained. However, this is likely due to the fact that both the pretest and post-test were measuring more than one latent variable and, therefore, the internal consistency reliability may have underestimated the “true” reliability or test-retest reliability.
Assessment Instruments

The following pretest and post-test instruments related to problem solving and BASIC language programming were employed for assessing the effect of different approaches to teach problem-solving skill on BASIC language programming ability.

1. The problem-solving pretest and post-test both consisted of 20 questions designed to measure the learners' problem-solving ability. A sample item is "Ben can never tell a lie. George can never tell the truth." Students are asked to detect who said what. A statement is made and one of the two identifies himself. Students must then decide who made the statement. Another example is the "Tower of Hanoi". This task assesses the problem-solving skills of the student. It has been widely studied in the problem solving area because it is basically a physical puzzle involving a series of observable steps which can be quantified and analyzed. The puzzle consists of three pegs. On one peg are arranged a number of disks of increasing size from top to bottom. The aim of the task is to transfer all the disks from the first peg to the third peg, always maintaining the order of small to large on a peg, with a minimum number of moves. The scoring simply involves determining the percentage of successful paths (minimal solution) at any given level.

2. BASIC language programming pretest and post-test - The pretest contains 15 questions and the post-test contains 20 questions which includes 12 BASIC
language understanding questions and 8 BASIC language program design questions.

Variables of the Study

The following dependent, covariate and independent variables were examined and studied.

Independent variables

The independent variables in this study included: (a) gender (male, female); and (b) group classification (control, induction, deduction).

Dependent variables

The dependent variables included: (a) post-test score in problem solving (PS-post), (b) the scores of two midterm exams in programming in the BASIC language; and (c) post-test score in programming in the BASIC language (BASIC-post).

Covariate Variables

The covariate variables included: (a) the pretest score in problem solving; and (b) the pretest score in programming in the BASIC language.
Experimental Treatments

Laboratory equipment

The laboratory equipment used in this study consisted of 50 IBM-PC compatible computers (i.e., Intel 486, 66 MHz CPU, using DOS 6.22 and Windows 3.1, Chinese version.)

Control group equipment - The control (wordprocessing) group in this study used the following items in the computer laboratory:
1. Microsoft QuickBASIC language (Chinese 1.5 version); and
2. Microsoft Word (Chinese 5.0c version).

Experimental group equipment - The two experimental (problem-solving method) groups in this study used only the Microsoft QuickBASIC language (Chinese 1.5 version).

Instructional materials

The fundamental instructional materials came from the textbook, which are interactive, self-instructional modules on computer programming in BASIC language programming. The problem-solving instruction used materials which employed heuristic strategies to develop problem-solving skills. There were also instructional materials relating to deductive and inductive problem-solving skills presented.

The BASIC computer language programming modules consisted of 20 print-based, self-instructional tutorials on BASIC language programming (see Appendix
F). Examples of the BASIC modules included: (a) an initial BASIC tutorial on the operations of the microcomputer system; (b) a guide to starting BASIC on the PC, and (c) a guide on Writing a Program—a description of line numbering, running, and listing in the BASIC program. The BASIC language modules were designed to teach the syntax or command structure of this language. Other lessons concentrated on applying procedures in programming design activities.

The problem-solving materials included 10 self-instructional tutorial modules on problem-solving strategies. Examples of these modules included: (a) Understand the Problem, which encourages completely understanding the problem; and (b) New Idea, which focuses on changing perspectives to solve problems (see Appendix D).

Experimental group

The three classes were randomly assigned to three treatments (control, deduction or induction) which were designed to compare groups that learned problem-solving methods with a group that did not learn problem solving on subsequent learning to program in the BASIC language.

Control group - The first group first received wordprocessing instruction in Word 5.0c for Windows followed by instruction in BASIC language programming instruction.
Deduction group - The second group first received three weeks of problem-solving skill learning by heuristics. Then, they had instruction using deduction in problem solving, followed by instruction in BASIC language programming.

Induction group - The last group first received three weeks of problem-solving skill learning by heuristics, then, instruction using induction to solve problems followed by instruction in BASIC language programming.

Hypotheses of the Study

Based on the questions of this study, sixteen general research hypotheses were formulated. The first four examined the effects of random assignment of sections to the experimental and control groups. Hypotheses 5 through 8 examined the effects of the students' achievement on the midterm tests in BASIC language concepts. Hypotheses 9 through 14 examined the effects of the students' achievement on the post-tests in BASIC language concepts and their problem-solving ability. Hypotheses 15 through 16 examined the correlation between the pretest and post-test achievement in BASIC language concepts and problem-solving ability.

Hypothesis 1: There is no difference between male and female pretest scores on problem solving beyond that which is expected due to random sampling variability tested at the 95% confidence level.

Ho: \( \mu_{ps\text{-pre, male}} = \mu_{ps\text{-pre, female}} \)
Hypothesis 2: There is no difference in the pretest scores on problem solving among students who receive problem-solving method one, problem-solving method two and wordprocessing beyond that which is expected due to random sampling variability tested at the 95% confidence level.

\[ H_0 : \mu_{\text{ps-pre, ps-method1}} = \mu_{\text{ps-pre, ps-method2}} = \mu_{\text{ps-pre, control}} \]

\[ H_a : \mu_{\text{ps-pre, ps-method1}} \neq \mu_{\text{ps-pre, ps-method2}} \neq \mu_{\text{ps-pre, control}} \]

Hypothesis 2.1: There is no difference in the pretest scores on problem solving among female students who receive problem-solving method one, problem-solving method two and wordprocessing beyond that which is expected due to random sampling variability tested at the 95% confidence level.

\[ H_0 : \mu_{\text{ps-pre, ps-method1,f}} = \mu_{\text{ps-pre, ps-method2,f}} = \mu_{\text{ps-pre, control,f}} \]

\[ H_a : \mu_{\text{ps-pre, ps-method1,f}} \neq \mu_{\text{ps-pre, ps-method2,f}} \neq \mu_{\text{ps-pre, control,f}} \]

Hypothesis 2.2: There is no difference in the pretest scores on problem solving between male students who receive problem-solving method two and wordprocessing beyond that which is expected due to random sampling variability tested at the 95% confidence level.

\[ H_0 : \mu_{\text{ps-pre, ps-method2,m}} = \mu_{\text{ps-pre, control,m}} \]

\[ H_a : \mu_{\text{ps-pre, ps-method2,m}} \neq \mu_{\text{ps-pre, control,m}} \]
Hypothesis 3: There is no difference between male and female pretest scores in learning problems in the BASIC language beyond that which is expected due to random sampling variability tested at the 95% confidence level.

\[ H_0: \mu_{\text{basic-pre, male}} = \mu_{\text{basic-pre, female}} \]

\[ H_a: \mu_{\text{basic-pre, male}} \neq \mu_{\text{basic-pre, female}} \]

Hypothesis 4: There is no difference in the pretest scores in learning to program in the BASIC language among students who receive problem-solving method one, problem-solving method two, and wordprocessing other than that which is expected due to random sampling variability tested at the 95% confidence level.

\[ H_0: \mu_{\text{basic-pre, ps-method1}} = \mu_{\text{basic-pre, ps-method2}} = \mu_{\text{basic-pre, control}} \]

\[ H_a: \mu_{\text{basic-pre, ps-method1}} \neq \mu_{\text{basic-pre, ps-method2}} \neq \mu_{\text{basic-pre, control}} \]

Hypothesis 4.1: There is no difference in the pretest scores in learning to program in the BASIC language among female students who receive problem-solving method one, problem-solving method two, and wordprocessing other than that which is expected due to random sampling variability tested at the 95% confidence level.

\[ H_0: \mu_{\text{basic-pre, ps-method1, f}} = \mu_{\text{basic-pre, ps-method2, f}} = \mu_{\text{basic-pre, control, f}} \]

\[ H_a: \mu_{\text{basic-pre, ps-method1, f}} \neq \mu_{\text{basic-pre, ps-method2, f}} \neq \mu_{\text{basic-pre, control, f}} \]

Hypothesis 4.2: There is no difference in the pretest scores in learning to program in the BASIC language between male students who receive problem-
solving method two, and wordprocessing other than that which is expected due to random sampling variability tested at the 95% confidence level.

\[ H_0 : \mu_{\text{basic-pre, ps-method2,m}} = \mu_{\text{basic-pre, control,m}} \]

\[ H_a : \mu_{\text{basic-pre, ps-method2,m}} \neq \mu_{\text{basic-pre, control,m}} \]

**Hypothesis 5:** There is no difference between male and female adjusted midterm exam one scores in learning to program in the BASIC language beyond that which is expected due to random sampling variability tested at the 95% confidence level.

\[ H_0 : \mu_{\text{basic-male, male}} = \mu_{\text{basic-male, female}} \]

\[ H_a : \mu_{\text{basic-male, male}} \neq \mu_{\text{basic-male, female}} \]

**Hypothesis 6:** There is no difference in the midterm exam one scores in learning to program in the BASIC language among students who receive problem-solving method one, problem-solving method two, and wordprocessing other than that which is expected due to random sampling variability tested at the 95% confidence level.

\[ H_0 : \mu_{\text{basic-male, ps-method1}} = \mu_{\text{basic-male, ps-method2}} = \mu_{\text{basic-male, control}} \]

\[ H_a : \mu_{\text{basic-male, ps-method1}} \neq \mu_{\text{basic-male, ps-method2}} \neq \mu_{\text{basic-male, control}} \]

**Hypothesis 6.1:** There is no difference in the midterm exam one scores in learning to program in the BASIC language among female students who receive problem-solving method one, problem-solving method two, and wordprocessing
other than that which is expected due to random sampling variability tested at the 95% confidence level.

\( \text{H}_0: \mu_{\text{basic-me1, ps-method1,f}} = \mu_{\text{basic-me1, ps-method2,f}} = \mu_{\text{basic-me1, control,f}} \)

\( \text{H}_a: \mu_{\text{basic-me1, ps-method1,f}} \neq \mu_{\text{basic-me1, ps-method2,f}} \neq \mu_{\text{basic-me1, control,f}} \)

**Hypothesis 6.2:** There is no difference in the midterm exam one scores in learning to program in the BASIC language between male students who receive problem-solving method two, and wordprocessing other than that which is expected due to random sampling variability tested at the 95% confidence level.

\( \text{H}_0: \mu_{\text{basic-me1, ps-method2,m}} = \mu_{\text{basic-me1, control,m}} \)

\( \text{H}_a: \mu_{\text{basic-me1, ps-method2,m}} \neq \mu_{\text{basic-me1, control,m}} \)

**Hypothesis 7:** There is no difference between male and female midterm exam two scores in learning to program in the BASIC language beyond that which is expected due to random sampling variability tested at the 95% confidence level.

\( \text{H}_0: \mu_{\text{basic-met2, male}} = \mu_{\text{basic-met2, female}} \)

\( \text{H}_a: \mu_{\text{basic-met2, male}} \neq \mu_{\text{basic-met2, female}} \)

**Hypothesis 8:** There is no difference in the midterm exam two scores in learning to program in the BASIC language among students who receive problem-solving method one, problem-solving method two, and wordprocessing other than that which is expected due to random sampling variability tested at the 95% confidence level.

\( \text{H}_0: \mu_{\text{basic-met2, ps-method1}} = \mu_{\text{basic-met2, ps-method2}} = \mu_{\text{basic-met2, control}} \)
Hypothesis 8.1: There is no difference in the midterm exam two scores in learning to program in the BASIC language among female students who receive problem-solving method one, problem-solving method two, and wordprocessing other than that which is expected due to random sampling variability tested at the 95% confidence level.

\[ H_0 : \mu_{\text{basic}-\text{met}2, \text{ps}-\text{method}1, \text{f}} = \mu_{\text{basic}-\text{met}2, \text{ps}-\text{method}2, \text{f}} = \mu_{\text{basic}-\text{met}2, \text{control}, \text{f}} \]

\[ H_a : \mu_{\text{basic}-\text{met}2, \text{ps}-\text{method}1, \text{f}} \neq \mu_{\text{basic}-\text{met}2, \text{ps}-\text{method}2, \text{f}} \neq \mu_{\text{basic}-\text{met}2, \text{control}, \text{f}} \]

Hypothesis 8.2: There is no difference in the midterm exam two scores in learning to program in the BASIC language between male students who receive problem-solving method two, and wordprocessing other than that which is expected due to random sampling variability tested at the 95% confidence level.

\[ H_0 : \mu_{\text{basic}-\text{met}2, \text{ps}-\text{method}2, \text{m}} = \mu_{\text{basic}-\text{met}2, \text{control}, \text{m}} \]

\[ H_a : \mu_{\text{basic}-\text{met}2, \text{ps}-\text{method}2, \text{m}} \neq \mu_{\text{basic}-\text{met}2, \text{control}, \text{m}} \]

Hypothesis 9: There is no difference between male and female adjusted post-test scores on problem solving beyond that which is expected due to random sampling variability tested at the 95% confidence level.

\[ H_0 : \mu_{\text{ps}-\text{post}, \text{male}} = \mu_{\text{ps}-\text{post}, \text{female}} \]

\[ H_a : \mu_{\text{ps}-\text{post}, \text{male}} \neq \mu_{\text{ps}-\text{post}, \text{female}} \]

Hypothesis 10: There is no difference in the adjusted post-test scores on problem solving among students who receive problem-solving method one,
problem-solving method two and wordprocessing beyond that which is expected due to random sampling variability tested at the 95% confidence level.

\[ H_0: \mu_{ps\text{-}post, ps\text{-}method1} = \mu_{ps\text{-}post, ps\text{-}method2} = \mu_{ps\text{-}post, control} \]

\[ H_a: \mu_{ps\text{-}post, ps\text{-}method1} \neq \mu_{ps\text{-}post, ps\text{-}method2} \neq \mu_{ps\text{-}post, control} \]

**Hypothesis 10.1:** There is no difference in the adjusted post-test scores on problem solving among female students who receive problem-solving method one, problem-solving method two and wordprocessing beyond that which is expected due to random sampling variability tested at the 95% confidence level.

\[ H_0: \mu_{ps\text{-}post, ps\text{-}method1,f} = \mu_{ps\text{-}post, ps\text{-}method2,f} = \mu_{ps\text{-}post, control,f} \]

\[ H_a: \mu_{ps\text{-}post, ps\text{-}method1,f} \neq \mu_{ps\text{-}post, ps\text{-}method2,f} \neq \mu_{ps\text{-}post, control,f} \]

**Hypothesis 10.2:** There is no difference in the adjusted post-test scores on problem solving between male students who receive problem-solving method two and wordprocessing beyond that which is expected due to random sampling variability tested at the 95% confidence level.

\[ H_0: \mu_{ps\text{-}post, ps\text{-}method2,m} = \mu_{ps\text{-}post, control,m} \]

\[ H_a: \mu_{ps\text{-}post, ps\text{-}method2,m} \neq \mu_{ps\text{-}post, control,m} \]

**Hypothesis 11:** There is no difference between male and female adjusted post-test scores in learning to program in the BASIC language beyond that which is expected due to random sampling variability tested at the 95% confidence level.

\[ H_0: \mu_{basic\text{-}post, male} = \mu_{basic\text{-}post, female} \]

\[ H_a: \mu_{basic\text{-}post, male} \neq \mu_{basic\text{-}post, female} \]
Hypothesis 12: There is no difference in the adjusted post-test scores in learning to program in the BASIC language among students who receive problem-solving method one, problem-solving method two, and wordprocessing other than that which is expected due to random sampling variability tested at the 95% confidence level.

\[ \text{Ho: } \mu_{\text{basic-post, ps-method1}} = \mu_{\text{basic-post, ps-method2}} = \mu_{\text{basic-post, control}} \]

\[ \text{Ha: } \mu_{\text{basic-post, ps-method1}} \neq \mu_{\text{basic-post, ps-method2}} \neq \mu_{\text{basic-post, control}} \]

Hypothesis 12.1: There is no difference in the adjusted post-test scores in learning to program in the BASIC language among female students who receive problem-solving method one, problem-solving method two, and wordprocessing other than that which is expected due to random sampling variability tested at the 95% confidence level.

\[ \text{Ho: } \mu_{\text{basic-post, ps-method1,f}} = \mu_{\text{basic-post, ps-method2,f}} = \mu_{\text{basic-post, control,f}} \]

\[ \text{Ha: } \mu_{\text{basic-post, ps-method1,f}} \neq \mu_{\text{basic-post, ps-method2,f}} \neq \mu_{\text{basic-post, control,f}} \]

Hypothesis 12.2: There is no difference in the adjusted post-test scores in learning to program in the BASIC language between male students who receive problem-solving method two, and wordprocessing other than that which is expected due to random sampling variability tested at the 95% confidence level.

\[ \text{Ho: } \mu_{\text{basic-post, ps-method2,f}} = \mu_{\text{basic-post, control,f}} \]

\[ \text{Ha: } \mu_{\text{basic-post, ps-method2,f}} \neq \mu_{\text{basic-post, control,f}} \]
Hypothesis 13: There is no difference in the adjusted post-test scores in programming design in the BASIC language among students who receive problem-solving method one, problem-solving method two, and wordprocessing other than that which is expected due to random sampling variability tested at the 95% confidence level.

\[ H_0 : \mu_{\text{basic-post, ps-method1}} = \mu_{\text{basic-post, ps-method2}} = \mu_{\text{basic-post, control}} \]

\[ H_a : \mu_{\text{basic-post, ps-method1}} \neq \mu_{\text{basic-post, ps-method2}} \neq \mu_{\text{basic-post, control}} \]

Hypothesis 13.1: There is no difference in the adjusted post-test scores in programming design in the BASIC language among female students who receive problem-solving method one, problem-solving method two, and wordprocessing other than that which is expected due to random sampling variability tested at the 95% confidence level.

\[ H_0 : \mu_{\text{basic-post, ps-method1,f}} = \mu_{\text{basic-post, ps-method2,f}} = \mu_{\text{basic-post, control,f}} \]

\[ H_a : \mu_{\text{basic-post, ps-method1,f}} \neq \mu_{\text{basic-post, ps-method2,f}} \neq \mu_{\text{basic-post, control,f}} \]

Hypothesis 13.2: There is no difference in the adjusted post-test scores in programming design in the BASIC language between male students who receive problem-solving method two, and wordprocessing other than that which is expected due to random sampling variability tested at the 95% confidence level.

\[ H_0 : \mu_{\text{basic-post, ps-method2,m}} = \mu_{\text{basic-post, control,m}} \]

\[ H_a : \mu_{\text{basic-post, ps-method2,m}} \neq \mu_{\text{basic-post, control,m}} \]
Hypothesis 14: There is no difference in the adjusted post-test scores in program understanding in the BASIC language among students who receive problem-solving method one, problem-solving method two, and wordprocessing other than that which is expected due to random sampling variability test at the 95% confidence level.

\[ H_0 : \mu_{\text{basic-post, ps-method1}} = \mu_{\text{basic-post, ps-method2}} = \mu_{\text{basic-post, control}} \]

\[ H_a : \mu_{\text{basic-post, ps-method1}} \neq \mu_{\text{basic-post, ps-method2}} \neq \mu_{\text{basic-post, control}} \]

Hypothesis 14.1: There is no difference in the adjusted post-test scores in program understanding in the BASIC language among female students who receive problem-solving method one, problem-solving method two, and wordprocessing other than that which is expected due to random sampling variability test at the 95% confidence level.

\[ H_0 : \mu_{\text{basic-post, ps-method1,f}} = \mu_{\text{basic-post, ps-method2,f}} = \mu_{\text{basic-post, control,f}} \]

\[ H_a : \mu_{\text{basic-post, ps-method1,f}} \neq \mu_{\text{basic-post, ps-method2,f}} \neq \mu_{\text{basic-post, control,f}} \]

Hypothesis 14.2: There is no difference in the adjusted post-test scores in program understanding in the BASIC language between male students who receive problem-solving method two, and wordprocessing other than that which is expected due to random sampling variability tested at the 95% confidence level.

\[ H_0 : \mu_{\text{basic-post, ps-method2,m}} = \mu_{\text{basic-post, control,m}} \]

\[ H_a : \mu_{\text{basic-post, ps-method2,m}} \neq \mu_{\text{basic-post, control,m}} \]
Hypothesis 15: The correlation between pretest and post-test scores in learning to program in the BASIC language does not differ from zero beyond that which is expected due to a random sampling variability tested at the 95\% confidence level.

\[ H_0: \rho = 0 \] and
\[ H_a: \rho \neq 0 \]

Hypothesis 16: The correlation between pretest and post-test scores on problem solving does not differ from zero beyond that which is expected due to random sampling variability tested at the 95\% confidence level.

\[ H_0: \rho = 0 \] and
\[ H_a: \rho \neq 0 \]

**Data Collection and Analyses**

**Statistical treatment**

Statistical analyses were performed to determine the results of the experiment.

1. A reliability analysis (Cronbach's alpha) was conducted to establish the internal consistency of the instrument for problem solving and the BASIC language pretest and post-test.
2. A 1 x 3 analysis of variance (ANOVA) was used to compare the results of the problem-solving pretest (male vs. female and control, deduction, and induction groups).

3. A 1 x 3 ANOVA was used to compare the results of the BASIC language programming pretest (male vs. female and control, deduction, induction groups).

4. A 1 x 3 analysis of covariance (ANCOVA) was used to compare the adjusted results of the BASIC language programming midterm two test score (male vs. female and control, deduction, and induction groups).

5. A 1 x 3 ANCOVA was used to compare the adjusted results of the BASIC language programming post-test achievement scores.

6. A 1 x 3 ANCOVA was used to compare the adjusted results of the problem-solving post-test achievement scores (combining design and understanding subtest scores).

7. A 1 x 3 ANCOVA was used to compare the adjusted results of the BASIC language programming design post-test scores.

8. A 1 x 3 ANCOVA was used to compare the adjusted results of the BASIC language programming understanding post-test scores.

9. The Pearson product-moment correlation coefficient was used to measure the relationship between the problem-solving pretest and the post-test scores.
10. The Pearson product-moment correlation coefficient was used to measure the relationship between the BASIC language programming pretest and the post-test achievement scores.

**Analysis**

Reliability estimates were obtained using the reliability procedure in the Statistical Package for the Social Sciences (SPSS). A one-way, fixed-effects analysis of covariance procedure and a one-way, fixed-effects analysis of variance procedure from the Statistical Analysis System (SAS) package were used to test the hypotheses of the study. Null hypotheses were tested at the 95% confidence interval level ($\alpha = .05$). The estimate of Type II errors at this Type I level was based on the pilot-test results and specification of an effect size of .5 standard deviation.
CHAPTER IV. RESULTS

In this chapter, the results and findings are discussed. The following sections are presented: (a) Descriptive Statistics; (b) Inferential Statistics; and (c) Findings. Descriptive statistics describe the general characteristics of the sample such as mean, standard deviation, etc., while inferential statistics provide the answers to the research questions and hypotheses discussed in Chapter I and Chapter III.

Descriptive Statistics

Tables 4.1 through 4.8 present the means and standard deviations of the pretest, midterm, and post-test. Values are shown for subjects grouped by gender and treatments.

Table 4.1. Means and standard deviations for the BASIC language pretest

<table>
<thead>
<tr>
<th>Type</th>
<th>Variable</th>
<th>Gender</th>
<th>N</th>
<th>Mean</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experiment group one (Deduction)</td>
<td>Female</td>
<td>32</td>
<td>2.187</td>
<td>1.821</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>32</td>
<td>2.187</td>
<td>1.821</td>
<td></td>
</tr>
<tr>
<td>By group</td>
<td>Experiment group two (Induction)</td>
<td>Male</td>
<td>09</td>
<td>2.333</td>
<td>2.236</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Female</td>
<td>10</td>
<td>2.500</td>
<td>1.080</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total</td>
<td>19</td>
<td>2.421</td>
<td>1.677</td>
</tr>
<tr>
<td></td>
<td>Control group</td>
<td>Male</td>
<td>10</td>
<td>2.800</td>
<td>1.619</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Female</td>
<td>14</td>
<td>2.000</td>
<td>1.921</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total</td>
<td>24</td>
<td>2.333</td>
<td>1.809</td>
</tr>
<tr>
<td>By Gender</td>
<td>Male</td>
<td>19</td>
<td>2.578</td>
<td>1.894</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Female</td>
<td>56</td>
<td>2.196</td>
<td>1.720</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>75</td>
<td>2.293</td>
<td>1.761</td>
<td></td>
</tr>
</tbody>
</table>

Range = 0 - 7
Table 4.2. Means and standard deviations for the problem-solving pretest

<table>
<thead>
<tr>
<th>Type</th>
<th>Variable</th>
<th>Gender</th>
<th>N</th>
<th>Mean</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experiment group one</td>
<td>Female</td>
<td>32</td>
<td>9.625</td>
<td>3.328</td>
<td></td>
</tr>
<tr>
<td>(deduction)</td>
<td>Total</td>
<td>32</td>
<td>9.625</td>
<td>3.328</td>
<td></td>
</tr>
<tr>
<td>By group</td>
<td>Experiment group two</td>
<td>Male</td>
<td>09</td>
<td>10.66</td>
<td>2.121</td>
</tr>
<tr>
<td>(induction)</td>
<td>Female</td>
<td>10</td>
<td>9.700</td>
<td>2.002</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>19</td>
<td>10.15</td>
<td>2.061</td>
<td></td>
</tr>
<tr>
<td>Control group</td>
<td>Male</td>
<td>10</td>
<td>10.20</td>
<td>2.347</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Female</td>
<td>14</td>
<td>9.700</td>
<td>2.064</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>24</td>
<td>10.41</td>
<td>2.145</td>
<td></td>
</tr>
<tr>
<td>By Gender Male</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Male</td>
<td>19</td>
<td>10.42</td>
<td>2.193</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Female</td>
<td>56</td>
<td>9.875</td>
<td>2.841</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>75</td>
<td>10.01</td>
<td>2.688</td>
<td></td>
</tr>
</tbody>
</table>

Range = 3 - 15

Table 4.3. Means and standard deviations for the midterm test one on BASIC language programming

<table>
<thead>
<tr>
<th>Type</th>
<th>Variable</th>
<th>Gender</th>
<th>N</th>
<th>Mean</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experiment group one</td>
<td>Female</td>
<td>32</td>
<td>.9437</td>
<td>.0913</td>
<td></td>
</tr>
<tr>
<td>(deduction)</td>
<td>Total</td>
<td>32</td>
<td>.9437</td>
<td>.0913</td>
<td></td>
</tr>
<tr>
<td>By group</td>
<td>Experiment group two</td>
<td>Male</td>
<td>09</td>
<td>.9111</td>
<td>.1054</td>
</tr>
<tr>
<td>(induction)</td>
<td>Female</td>
<td>10</td>
<td>.8700</td>
<td>.1494</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>19</td>
<td>.8894</td>
<td>.1286</td>
<td></td>
</tr>
<tr>
<td>Control group</td>
<td>Male</td>
<td>10</td>
<td>.6000</td>
<td>.5163</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Female</td>
<td>14</td>
<td>.6214</td>
<td>.2965</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>24</td>
<td>.7291</td>
<td>.4080</td>
<td></td>
</tr>
<tr>
<td>By Gender Male</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Male</td>
<td>19</td>
<td>.7473</td>
<td>.4046</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Female</td>
<td>56</td>
<td>.9000</td>
<td>.1788</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>75</td>
<td>.8613</td>
<td>.2609</td>
<td></td>
</tr>
</tbody>
</table>

Range = 0 - 1
Table 4.4. Means and standard deviations for midterm test two on BASIC language programming

<table>
<thead>
<tr>
<th>Type</th>
<th>Variable</th>
<th>Gender</th>
<th>N</th>
<th>Mean</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experiment group one</td>
<td>Female</td>
<td>32</td>
<td>72.06</td>
<td>11.26</td>
<td></td>
</tr>
<tr>
<td>(deduction)</td>
<td>Total</td>
<td>32</td>
<td>72.06</td>
<td>11.26</td>
<td></td>
</tr>
<tr>
<td>By group</td>
<td>Experiment group two</td>
<td>Male</td>
<td>09</td>
<td>72.77</td>
<td>16.60</td>
</tr>
<tr>
<td>(induction)</td>
<td>Female</td>
<td>10</td>
<td>82.50</td>
<td>13.59</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>19</td>
<td>77.89</td>
<td>15.48</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Control group</td>
<td>Male</td>
<td>10</td>
<td>47.00</td>
<td>22.38</td>
</tr>
<tr>
<td></td>
<td>Female</td>
<td>14</td>
<td>37.35</td>
<td>17.65</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>24</td>
<td>41.37</td>
<td>19.89</td>
<td></td>
</tr>
<tr>
<td>By Gender</td>
<td>Male</td>
<td>19</td>
<td>59.21</td>
<td>23.40</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Female</td>
<td>56</td>
<td>65.25</td>
<td>21.31</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>75</td>
<td>63.72</td>
<td>21.86</td>
<td></td>
</tr>
</tbody>
</table>

Range = 8 -100

Table 4.5. Means and standard deviations for the BASIC language post-test

<table>
<thead>
<tr>
<th>Type</th>
<th>Variable</th>
<th>Gender</th>
<th>N</th>
<th>Mean</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experiment group one</td>
<td>Female</td>
<td>32</td>
<td>9.437</td>
<td>2.487</td>
<td></td>
</tr>
<tr>
<td>(deduction)</td>
<td>Total</td>
<td>32</td>
<td>9.437</td>
<td>2.487</td>
<td></td>
</tr>
<tr>
<td>By group</td>
<td>Experiment group two</td>
<td>Male</td>
<td>09</td>
<td>12.00</td>
<td>3.500</td>
</tr>
<tr>
<td>(induction)</td>
<td>Female</td>
<td>10</td>
<td>11.70</td>
<td>1.702</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>19</td>
<td>11.84</td>
<td>2.630</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Control group</td>
<td>Male</td>
<td>10</td>
<td>6.000</td>
<td>5.163</td>
</tr>
<tr>
<td></td>
<td>Female</td>
<td>14</td>
<td>6.071</td>
<td>2.464</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>24</td>
<td>6.541</td>
<td>2.797</td>
<td></td>
</tr>
<tr>
<td>By Gender</td>
<td>Male</td>
<td>19</td>
<td>9.473</td>
<td>4.087</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Female</td>
<td>56</td>
<td>9.000</td>
<td>3.003</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>75</td>
<td>9.120</td>
<td>3.287</td>
<td></td>
</tr>
</tbody>
</table>

Range = 3 - 17
### Table 4.6. Means and standard deviations for the problem-solving post-test

<table>
<thead>
<tr>
<th>Type</th>
<th>Variable</th>
<th>Gender</th>
<th>N</th>
<th>Mean</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experiment group one</td>
<td>(deduction)</td>
<td>Female</td>
<td>32</td>
<td>14.25</td>
<td>2.257</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total</td>
<td>32</td>
<td>14.25</td>
<td>2.257</td>
</tr>
<tr>
<td>By group</td>
<td></td>
<td>Male</td>
<td>9</td>
<td>15.11</td>
<td>1.536</td>
</tr>
<tr>
<td>Control group</td>
<td></td>
<td>Female</td>
<td>10</td>
<td>15.20</td>
<td>1.751</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total</td>
<td>19</td>
<td>15.15</td>
<td>1.607</td>
</tr>
<tr>
<td>By Gender</td>
<td>Male</td>
<td>19</td>
<td>14.26</td>
<td>1.851</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Female</td>
<td>56</td>
<td>14.12</td>
<td>2.123</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>75</td>
<td>14.16</td>
<td>2.047</td>
<td></td>
</tr>
</tbody>
</table>

Range = 9.19

### Table 4.7. Means and standard deviations for the BASIC language post-test in program design

<table>
<thead>
<tr>
<th>Type</th>
<th>Variable</th>
<th>Gender</th>
<th>N</th>
<th>Mean</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experiment group one</td>
<td>(deduction)</td>
<td>Female</td>
<td>32</td>
<td>4.468</td>
<td>1.243</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total</td>
<td>32</td>
<td>4.468</td>
<td>1.243</td>
</tr>
<tr>
<td>By group</td>
<td></td>
<td>Male</td>
<td>9</td>
<td>4.555</td>
<td>1.424</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Female</td>
<td>10</td>
<td>4.600</td>
<td>0.699</td>
</tr>
<tr>
<td>Control group</td>
<td></td>
<td>Male</td>
<td>10</td>
<td>2.800</td>
<td>1.229</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Female</td>
<td>14</td>
<td>2.571</td>
<td>1.741</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total</td>
<td>24</td>
<td>2.666</td>
<td>1.522</td>
</tr>
<tr>
<td>By Gender</td>
<td>Male</td>
<td>19</td>
<td>3.631</td>
<td>1.570</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Female</td>
<td>56</td>
<td>4.017</td>
<td>1.543</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>75</td>
<td>3.920</td>
<td>1.548</td>
<td></td>
</tr>
</tbody>
</table>

Range = 0 - 7
Table 4.8. Means and standard deviations for the BASIC language post-test in program understanding

<table>
<thead>
<tr>
<th>Type</th>
<th>Variable</th>
<th>Gender</th>
<th>N</th>
<th>Mean</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>By group</td>
<td>Experiment group one</td>
<td>Female</td>
<td>32</td>
<td>4.968</td>
<td>1.731</td>
</tr>
<tr>
<td>(deduction)</td>
<td>Total</td>
<td></td>
<td>32</td>
<td>4.968</td>
<td>1.731</td>
</tr>
<tr>
<td>Experiment group two</td>
<td>Male</td>
<td>09</td>
<td>7.444</td>
<td>2.554</td>
<td></td>
</tr>
<tr>
<td>(induction)</td>
<td>Female</td>
<td>10</td>
<td>7.100</td>
<td>1.911</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>19</td>
<td>7.263</td>
<td>2.181</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control group</td>
<td>Male</td>
<td>10</td>
<td>4.400</td>
<td>2.366</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Female</td>
<td>14</td>
<td>3.500</td>
<td>1.400</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>24</td>
<td>3.875</td>
<td>1.872</td>
<td></td>
</tr>
<tr>
<td>By Gender</td>
<td>Male</td>
<td>19</td>
<td>5.842</td>
<td>2.853</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Female</td>
<td>56</td>
<td>4.982</td>
<td>2.031</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>75</td>
<td>5.200</td>
<td>2.277</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Range = 1 - 10

Inferential Statistics

This section presents the results of tests on the research hypotheses. First reported are the pretest hypotheses (1 - 4.2), midterm test hypotheses (5 - 8.2), post-test hypotheses (9 - 12.2), detail BASIC language post-test in design and understanding hypotheses (13 - 14.2), and correlation hypotheses (15 - 16).

Pretest hypotheses

From the data shown in Table 4.9, the critical value for rejection of the null hypotheses was based on the probability of a larger F statistic by chance alone as being less than 0.05. Therefore, Hypotheses 1 through 4.2 were retained. It was
Table 4.9. Summary of one-way ANOVA for pretest mean scores

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Measure</th>
<th>Group</th>
<th>F-Statistic</th>
<th>Prob &gt; F</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>problem-solving pretest</td>
<td>M, F</td>
<td>0.58</td>
<td>0.45</td>
<td>Not Sig.</td>
</tr>
<tr>
<td>2</td>
<td>problem-solving pretest</td>
<td>I,D,W</td>
<td>0.62</td>
<td>0.54</td>
<td>Not Sig.</td>
</tr>
<tr>
<td>2.1</td>
<td>problem-solving pretest</td>
<td>I-F,D-F,W-F</td>
<td>0.55</td>
<td>0.58</td>
<td>Not Sig.</td>
</tr>
<tr>
<td>2.2</td>
<td>problem-solving pretest</td>
<td>I-M, W-M</td>
<td>0.20</td>
<td>0.66</td>
<td>Not Sig.</td>
</tr>
<tr>
<td>3</td>
<td>BASIC concepts pretest</td>
<td>M, F</td>
<td>0.67</td>
<td>0.42</td>
<td>Not Sig.</td>
</tr>
<tr>
<td>4</td>
<td>BASIC concepts pretest</td>
<td>I,D,W</td>
<td>0.11</td>
<td>0.89</td>
<td>Not Sig.</td>
</tr>
<tr>
<td>4.1</td>
<td>BASIC concepts pretest</td>
<td>I-F,D-F,W-F</td>
<td>0.24</td>
<td>0.79</td>
<td>Not Sig.</td>
</tr>
<tr>
<td>4.2</td>
<td>BASIC concepts pretest</td>
<td>I-M, W-M</td>
<td>0.28</td>
<td>0.61</td>
<td>Not Sig.</td>
</tr>
</tbody>
</table>

M = male group; F = female group  
I = induction group; D = deduction group; W = word process group  
I-F = induction female group; D-F = deduction female group; W-F = word process female group; I-M = induction male group; W-M = word process male group

concluded that there is no difference in the two pretests for any of the planned comparisons.

Midterm hypotheses

From the data shown in Table 4.10, the probability of a larger F statistic for Hypothesis 5 was 0.037, therefore Hypothesis 5 was rejected. It was concluded that there is a significant difference between the male and female adjusted midterm exam one scores in learning the BASIC language. The male group mean was 7.47 whereas the female group mean was 9.00.
Table 4.10. Summary of the ANCOVAs on the midterm examinations

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Measure</th>
<th>Group</th>
<th>F-Statistic</th>
<th>Prob &gt; F</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>BASIC midterm one</td>
<td>M, F</td>
<td>4.50</td>
<td>0.037</td>
<td>Sig.</td>
</tr>
<tr>
<td>6</td>
<td>BASIC midterm one</td>
<td>I,D,W</td>
<td>5.16</td>
<td>0.008</td>
<td>Sig.</td>
</tr>
<tr>
<td>6.1</td>
<td>BASIC midterm one</td>
<td>I-F,D-F,W-F</td>
<td>2.26</td>
<td>0.114</td>
<td>Not Sig.</td>
</tr>
<tr>
<td>6.2</td>
<td>BASIC midterm one</td>
<td>I-M, W-M</td>
<td>2.70</td>
<td>0.127</td>
<td>Not Sig.</td>
</tr>
<tr>
<td>7</td>
<td>BASIC midterm two</td>
<td>M, F</td>
<td>1.30</td>
<td>0.258</td>
<td>Not Sig.</td>
</tr>
<tr>
<td>8</td>
<td>BASIC midterm two</td>
<td>I,D,W</td>
<td>38.2</td>
<td>&lt;0.001</td>
<td>Sig.</td>
</tr>
<tr>
<td>8.1</td>
<td>BASIC midterm two</td>
<td>I-F,D-F,W-F</td>
<td>39.2</td>
<td>&lt;0.001</td>
<td>Sig.</td>
</tr>
<tr>
<td>8.2</td>
<td>BASIC midterm two</td>
<td>I-M, W-M</td>
<td>11.5</td>
<td>0.004</td>
<td>Sig.</td>
</tr>
</tbody>
</table>

M = male group; F = female group
I = induction group; D = deduction group; W= word process group
I-F = induction female group; D-F = deduction female group; W-F = word process female group; I-M = induction male group; W-M = word process male group

For Hypothesis 6, the probability of a larger F statistic was 0.008, therefore Hypothesis 6 was rejected. It was concluded that there was a significant difference in the BASIC language midterm one scores among problem-solving one (deduction group), problem-solving two (induction group), and wordprocessing (control group). Therefore, the Scheffe post-hoc multiple range test was used to test the significant differences among the group levels. The results of Scheffe's test are shown in Table 4.11. There were significant differences between the control group and deduction group.
Table 4.11. Results of midterm test one scores

<table>
<thead>
<tr>
<th>Mean</th>
<th>Group</th>
<th>Deduction</th>
<th>Induction</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.437</td>
<td>deduction</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.894</td>
<td>induction</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.291</td>
<td>control</td>
<td>*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Significantly different at the 0.05 level

deduction = problem-solving group one
induction = problem-solving group two

Based on the small chance probability for the F statistics obtained for Hypotheses 8, 8.1 and 8.2, they were all rejected. It was concluded that there was a significant difference in the two BASIC language midterm scores among the problem-solving one, problem-solving two, and wordprocessing groups. The Scheffe post-hoc multiple range test was used to determine the significant differences among group levels. The deduction and induction group midterm test two scores were both higher than the control group midterm test two scores in female students. The induction male students midterm test two scores were higher than the control male students midterm test two scores. The results of Scheffe's test are shown in Table 4.12, 4.13, and 4.14.
Table 4.12. Results of the midterm test two scores

<table>
<thead>
<tr>
<th>Mean</th>
<th>Group</th>
<th>Deduction</th>
<th>Induction</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>72.06</td>
<td>deduction</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>77.89</td>
<td>induction</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41.37</td>
<td>control</td>
<td>*</td>
<td>*</td>
<td></td>
</tr>
</tbody>
</table>

* Significantly different at the 0.05 level
deduction = problem-solving group one
induction = problem-solving group two

Table 4.13. Results of midterm test two scores for the female group

<table>
<thead>
<tr>
<th>Mean</th>
<th>Group</th>
<th>Deduction</th>
<th>Induction</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>72.06</td>
<td>deduction</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>82.50</td>
<td>induction</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37.25</td>
<td>control</td>
<td>*</td>
<td>*</td>
<td></td>
</tr>
</tbody>
</table>

* Significantly different at the 0.05 level
deduction = problem-solving group one
induction = problem-solving group two

Table 4.14. Results of midterm test two scores for the male group

<table>
<thead>
<tr>
<th>Mean</th>
<th>Group</th>
<th>Induction</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>72.77</td>
<td>induction</td>
<td></td>
<td></td>
</tr>
<tr>
<td>47.00</td>
<td>control</td>
<td>*</td>
<td></td>
</tr>
</tbody>
</table>

* Significantly different at the 0.05 level
induction = problem-solving group two
Post-test hypotheses

Based on the F statistics for Hypotheses 10, and 10.1 as shown in Table 4.15, these two hypotheses were rejected. It was concluded that there was a significant difference in the problem-solving post-test scores among problem-solving one, problem-solving two, and wordprocessing groups. Therefore, the Scheffe post-hoc multiple range test was used to test the significant differences among group levels. The results of Scheffe’s test are shown in Table 4.16 and 4.17.

Table 4.15. Summary of the ANCOVAs for the post-test examinations

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Measure</th>
<th>Group</th>
<th>F-Statistic</th>
<th>Prob &gt; F</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>problem-solving post-test</td>
<td>M, F</td>
<td>0.03</td>
<td>0.855</td>
<td>Not Sig.</td>
</tr>
<tr>
<td>10</td>
<td>problem-solving post-test</td>
<td>I,D,W</td>
<td>8.41</td>
<td>&lt;0.001</td>
<td>Sig.</td>
</tr>
<tr>
<td>10.1</td>
<td>problem-solving post-test</td>
<td>I-F,D-F,W-F</td>
<td>7.83</td>
<td>0.0011</td>
<td>Sig.</td>
</tr>
<tr>
<td>10.2</td>
<td>problem-solving post-test</td>
<td>I-M, W-M</td>
<td>3.54</td>
<td>0.079</td>
<td>Not Sig.</td>
</tr>
<tr>
<td>11</td>
<td>BASIC post-test</td>
<td>M, F</td>
<td>0.01</td>
<td>0.903</td>
<td>Not Sig.</td>
</tr>
<tr>
<td>12</td>
<td>BASIC post-test</td>
<td>I,D,W</td>
<td>34.4</td>
<td>&lt;0.001</td>
<td>Sig.</td>
</tr>
<tr>
<td>12.1</td>
<td>BASIC post-test</td>
<td>I-F,D-F,W-F</td>
<td>24.2</td>
<td>&lt;0.001</td>
<td>Sig.</td>
</tr>
<tr>
<td>12.2</td>
<td>BASIC post-test</td>
<td>I-M, W-M</td>
<td>17.7</td>
<td>&lt;0.001</td>
<td>Sig.</td>
</tr>
</tbody>
</table>

M = male group; F = female group
I = induction group; D = deduction group; W = wordprocessing group
I-F = induction female group; D-F = deduction female group; W-F = wordprocessing female group; I-M = induction male group; W-M = wordprocessing male group.
Table 4.16. Problem-solving post-test scores

<table>
<thead>
<tr>
<th>Mean</th>
<th>Group</th>
<th>Deduction</th>
<th>Induction</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>14.25</td>
<td>deduction</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15.15</td>
<td>induction</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13.25</td>
<td>control</td>
<td></td>
<td></td>
<td>*</td>
</tr>
</tbody>
</table>

* Significantly different at the 0.05 level

deduction = problem-solving group one
induction = mean problem-solving group two

Table 4.17. Problem-solving post-test scores for the female group

<table>
<thead>
<tr>
<th>Mean</th>
<th>Group</th>
<th>Deduction</th>
<th>Induction</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>14.25</td>
<td>deduction</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15.20</td>
<td>induction</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13.07</td>
<td>control</td>
<td></td>
<td></td>
<td>*</td>
</tr>
</tbody>
</table>

* Significantly different at the 0.05 level

deduction = problem-solving group one
induction = problem-solving group two

Based on the small chance probability for the F statistics obtained for Hypotheses 12, 12.1 and 12.2, they were all rejected. It was concluded that there is a significant difference in the BASIC language post-test scores among problem-solving one, problem-solving two, and wordprocessing groups. The Scheffe post-hoc multiple range test was used to test the significant differences among group levels. The results of the Scheffe's test are shown in Table 4.18, 4.19, and 4.20.
Table 4.18 BASIC language post-test scores

<table>
<thead>
<tr>
<th>Mean</th>
<th>Group</th>
<th>Deduction</th>
<th>Induction</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.437</td>
<td>deduction</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11.84</td>
<td>induction</td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.541</td>
<td>control</td>
<td>*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Significantly different at the 0.05 level
deduction = problem-solving group one
induction = problem-solving group two

Table 4.19. BASIC language post-test scores for the female group

<table>
<thead>
<tr>
<th>Mean</th>
<th>Group</th>
<th>Deduction</th>
<th>Induction</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.437</td>
<td>deduction</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11.70</td>
<td>induction</td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.071</td>
<td>control</td>
<td>*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Significantly different at the 0.05 level
deduction = problem-solving group one
induction = problem-solving group two

Table 4.20. BASIC language post-test scores for the male group

<table>
<thead>
<tr>
<th>Mean</th>
<th>Group</th>
<th>Induction</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.00</td>
<td>induction</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.200</td>
<td>control</td>
<td></td>
<td>*</td>
</tr>
</tbody>
</table>

* Significantly different at the 0.05 level
induction = problem-solving group two
Hypotheses regarding BASIC post-tests in design and understanding

Based on the small probabilities corresponding to the F statistics obtained in Hypotheses 13, 13.1 and 13.2, as shown in Table 21, they were all rejected. It was concluded that there was a significant difference in the BASIC language post-test scores in design among problem-solving one, problem-solving two, and wordprocessing groups. The Scheffe post-hoc multiple range test was used to test the significant differences among the group. The results of the Scheffe's test are shown in Table 4.22, 4.23, and 4.24.

Table 4.21. Summary of the ANCOVAs for BASIC post-test in design and understanding

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Measure</th>
<th>Group</th>
<th>F-Statistic</th>
<th>Prob &gt; F</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>BASIC post-test design</td>
<td>I,D,W</td>
<td>24.6</td>
<td>&lt;0.001</td>
<td>Sig.</td>
</tr>
<tr>
<td>13.1</td>
<td>BASIC post-test design</td>
<td>I-F,D-F,W-F</td>
<td>14.9</td>
<td>&lt;0.001</td>
<td>Sig.</td>
</tr>
<tr>
<td>13.2</td>
<td>BASIC post-test design</td>
<td>I-M, W-M</td>
<td>11.7</td>
<td>0.001</td>
<td>Sig.</td>
</tr>
<tr>
<td>14</td>
<td>BASIC posttest understanding</td>
<td>I,D,W</td>
<td>21.9</td>
<td>&lt;0.001</td>
<td>Sig.</td>
</tr>
<tr>
<td>14.1</td>
<td>BASIC posttest understanding</td>
<td>I-F,D-F,W-F</td>
<td>13.7</td>
<td>&lt;0.001</td>
<td>Sig.</td>
</tr>
<tr>
<td>14.2</td>
<td>BASIC posttest understanding</td>
<td>I-M, W-M</td>
<td>13.53</td>
<td>0.002</td>
<td>Sig.</td>
</tr>
</tbody>
</table>

M = male group; F = female group
I = induction group; D = deduction group; W = wordprocessing group
I-F = induction female group; D-F = deduction female group; W-F = wordprocessing female group; I-M = induction male group; W-M = word process male group
### Table 4.22. BASIC language post-test scores in design

<table>
<thead>
<tr>
<th>Mean</th>
<th>Group</th>
<th>Deduction</th>
<th>Induction</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.468</td>
<td>deduction</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.578</td>
<td>induction</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.666</td>
<td>control</td>
<td>*</td>
<td></td>
<td>*</td>
</tr>
</tbody>
</table>

* Significantly different at the 0.05 level  
deduction = problem-solving group one  
induction = problem-solving group two

### Table 4.23. BASIC language post-test scores in design for the female group

<table>
<thead>
<tr>
<th>Mean</th>
<th>Group</th>
<th>Deduction</th>
<th>Induction</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.468</td>
<td>deduction</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.600</td>
<td>induction</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.571</td>
<td>control</td>
<td>*</td>
<td></td>
<td>*</td>
</tr>
</tbody>
</table>

* Significantly different at the 0.05 level  
deduction = problem-solving group one  
induction = problem-solving group two

### Table 4.24. BASIC language post-test scores in design scores for the male group

<table>
<thead>
<tr>
<th>Mean</th>
<th>Group</th>
<th>Induction</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.555</td>
<td>induction</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.800</td>
<td>control</td>
<td></td>
<td>*</td>
</tr>
</tbody>
</table>

* Significantly different at the 0.05 level  
induction = problem-solving group two
Based on the small probabilities for the F statistics obtained in Hypothesis 14, 14.1 and 14.2, they were all rejected. It was concluded that there were significant differences in the BASIC language in understanding post-test scores among problem-solving one, problem-solving two, and the wordprocessing groups. The Scheffe post-hoc multiple range test was used to determine the significant differences among the group levels. The results of the Scheffe's test are shown in Table 4.25, 4.26, and 4.27.

Table 4.25. BASIC language post-test scores in understanding

<table>
<thead>
<tr>
<th>Mean</th>
<th>Group</th>
<th>Deduction</th>
<th>Induction</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.968</td>
<td>deduction</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.263</td>
<td>induction</td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.875</td>
<td>control</td>
<td></td>
<td>*</td>
<td></td>
</tr>
</tbody>
</table>

* Significantly different at the 0.05 level
deduction = problem-solving group one
induction = problem-solving group two

Table 4.26. BASIC language post-test scores in understanding for the female group

<table>
<thead>
<tr>
<th>Mean</th>
<th>Group</th>
<th>Deduction</th>
<th>Induction</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.968</td>
<td>deduction</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.100</td>
<td>induction</td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.500</td>
<td>control</td>
<td></td>
<td>*</td>
<td></td>
</tr>
</tbody>
</table>

* Significantly different at the 0.05 level
deduction = problem-solving group one
induction = problem-solving group two
Table 4.27. BASIC language post-test scores in understanding for the male group

<table>
<thead>
<tr>
<th>Mean</th>
<th>Group</th>
<th>Induction</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.444</td>
<td>induction</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.400</td>
<td>control</td>
<td></td>
<td>*</td>
</tr>
</tbody>
</table>

* Significantly different at the 0.05 level
induction = problem-solving group two

**Correlation hypotheses**

Based on the F statistics for Hypotheses 15 and 16, as shown in Table 4.28, these two hypotheses were rejected. It was concluded that there was a significant correlation between the pretest and post-test in the BASIC language concepts and problem solving. The value of \( r = 0.465 \) and 0.488, respectively, indicates there was a positive relationship between the pretest and post-test in the BASIC language concepts and problem-solving tests.

Table 4.28. Summary of correlation between the BASIC language pretest and post-test; problem-solving pretest and post-test

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Measure</th>
<th>r-coefficient</th>
<th>Prob &gt; F</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>BASIC concepts</td>
<td>0.465</td>
<td>&lt;0.001</td>
<td>*</td>
</tr>
<tr>
<td>16</td>
<td>Problem-solving</td>
<td>0.488</td>
<td>&lt;0.001</td>
<td>*</td>
</tr>
</tbody>
</table>

* Significantly different at the 0.05 level
Findings

After an analysis of the data collected from the pretest, midterm, and post-test examinations, the following results were revealed:

1. No difference was found between male and female groups, among the three treatments in BASIC language concepts and problem solving on the pretest.

2. There was a significant difference on the BASIC midterm one test between male and female groups and the three treatment groups. There was also a significant difference on the BASIC midterm two test among (a) the total induction group, the total deduction group, and the total wordprocessing group; (b) the induction female group, the deduction female group, and the wordprocessing female group; and (c) the induction male group and the wordprocessing male group. The female students scored higher than the male students in the BASIC midterm one test, and the problem-solving method one (deduction) group scored higher than the non problem-solving group on the BASIC midterm one test. The two problem-solving method (deduction and induction) groups scored higher than the non problem-solving method group on the BASIC midterm two test than the female and total students groups. The induction male group BASIC midterm test two scores were higher than the wordprocessing male group.

3. There was a significant difference on the problem-solving post-test among the three treatment groups (Figure 4.1) and among the female students in the
three treatment groups. There was also a significant difference on the BASIC post-test among the three treatment groups (Figure 4.2), and in the male students between the control and induction groups (Figure 4.3) and the female students in the three treatment groups (Figure 4.4). The two problem-solving method groups BASIC post-test score higher than the non problem-solving method group BASIC post-test score in the female and total students. The induction male group post-test scored higher than the control male group post-test scored. The two problem-solving methods problem-solving post-test scored both higher than control group post-test scored.
Figure 4.2. Means for the BASIC language pretest and post-test

Figure 4.3. Means for the BASIC language pretest and post-test (male)
4. There was a significant difference on the BASIC post-test on design, and the understanding subtests among the three treatment groups, among the female students in the three treatment groups, and among the male students in the three treatment groups. The two problem-solving method groups scored higher on the BASIC post-test on design and understanding than the respective non problem-solving method group.

5. The female induction group scored higher than the female deduction group on the BASIC language post-test in understanding.

6. There were significant positive correlations between the pretest and post-test on problem solving and the BASIC language programming.
Summary

The results of the statistical analyses and the findings were presented in this chapter. The general characteristics of the pretest, midterm test one, midterm test two, and post-test result were explained using means and standard deviations. The results indicated that when female students first study problem-solving methods (induction and deduction) they experience a significant increase in BASIC language programming achievement. Likewise, male students who first learn problem solving (induction) experience a significant increase in BASIC language program achievement.

The study also showed that female students who first receive problem-solving instruction in induction subsequently learn BASIC language programming significantly better than female students who first receive problem-solving instruction in deduction and subsequently learn BASIC language programming.

Further evidence supports that female students in group one and two on BASIC language programming in design and understanding performed significantly better than the female students in the control group. In addition, the male students who first learned problem solving (induction) performed significantly better than the males who first received non-problem solving instruction prior to learning BASIC language programming in design and understanding.
CHAPTER V. SUMMARY, DISCUSSION, CONCLUSIONS, AND RECOMMENDATIONS

In the previous four chapters of this study, the introduction, review of literature, methodology, and statistical analysis and findings were presented. This chapter presents a summary of this study, provides discussion, and makes conclusions based on the findings and results. Finally, recommendations for future research are presented.

Summary

Learning problem solving is an important part of cognitive development. Teachers must promote the development of problem-solving skills in their students. If problem-solving skill development is actively and sequentially promoted in learning programming, it can also be applied in a related area.

The experimental research design in this study was conducted with 75 students enrolled in National Chiayi Teachers College in Taiwan during the spring of 1995. Sections of students were randomly assigned to either one of two experimental groups, or to a control group. The pretest scores in BASIC language concepts and problem solving showed that the randomly assigned groups were equal or nearly equal on these tests.
This research used a pretest/post-test control-group design (Campbell & Stanley, 1963). The three groups of students (two experimental groups and one control group) took six tests: two paper and pencil knowledge pretests, two midterm tests, and two post-tests. Homework was handed in weekly and two teacher-made midterm tests were held after the first two teachers completed BASIC language instruction. The midterm tests were composed of several of problems from homework and textbook assignments. The final exams on problem solving and on BASIC language programming covered all material taught in the BASIC programming course.

Comparisons were made among the three treatment groups, and the results were presented in Chapter IV. Comparisons were also made between gender and by treatments within gender. Data from the tests were analyzed using the ANOVA and ANCOVA to determine if statistically significant differences existed among the groups. The level of significance was set at $\alpha = 0.05$.

The results of this research showed there were significant differences in the means of the BASIC language programming scores among the three treatments and between genders on midterm test one. Significant differences were found among the three groups in BASIC language midterm test two, BASIC post-test, post-test in programming design and post-test in program understanding by the male group, female group, and the combined group.
The results of this research showed that were significant differences in the mean ratings of the post-test problem-solving scores by the female group and the combined group among three treatments. There were significant positive correlations between the pretest and post-test in problem solving and BASIC language concepts.

Discussion

The purpose of this study was to compare learning problem-solving methods versus a controlled, non problem-solving activity such as word-processing with subsequent learning to program in the BASIC language. From the results of the statistical analyses, several factors can be discussed.

1. A comparison of the performance can be made between male and female students on the BASIC language programming midterm test one. From the results, it can be inferred that females performed better than males on this test. Learning BASIC programming first requires one to have an ability to understand language commands, and the female students in the study scored better on midterm test one than the male students. Matlin et al. (1985) reported that females are stronger than males in the verbal domain which could support this result. In addition, the deduction group, which only contained female students, scored higher on midterm one test than the
control group which was comprised of both male and female students (see Table 4.10).

2. There was not enough evidence to reject Hypotheses 5 and 6, that compared the performance of female and male students independently of one another among the three experiment groups on BASIC language programming midterm one. This might be due to the fact that midterm one contained only one question, which hardly determines acquired learning. Another reason may be that students usually do not have difficulty with learning BASIC programming first, which does not require the flow of control and procedure. Thus, this BASIC programming performance was equally easy for the three groups.

3. There was not enough evidence to reject Hypotheses 7 and 11, that compared the performance between male and female students on the BASIC language programming midterm two test and the post-test. It is possible that the instruction in deduction and induction had equally affected both males and females on their skills as applied to BASIC programming. Of course, it may be that males and females were simply equal in programming and neither group was affected to a greater or lesser degree by the pre-instruction.

4. When comparing the three experiment groups, statistically significant differences were found among the three treatments on BASIC programming
midterm test two, and the post-test. The two problem-solving method groups scored higher than the non problem-solving group on midterm test two and the post-test. It might be that students who received deduction or induction with problem-solving methods instruction had fewer logic and syntax errors and had a greater understanding of BASIC programming flow of control than students who received non problem-solving methods instruction.

5. The female induction group scored better than the female deduction group on the BASIC language post-test in understanding. It might be that female students who learned the inductive process developed new concepts using domain knowledge which helped them to achieve higher scores. The inductive process may have enabled them to become more intuitive and apply creative thought as a skill along with their personal knowledge. The inductive process may broaden one’s abilities to understand general programming concepts as opposed to simply learning BASIC programming.

6. The two problem-solving methods groups (induction and deduction) scored higher than the non problem-solving group on the BASIC post-test on design and understanding. It is possible that both deduction and induction may enable students to express themselves more creatively in finding solutions to programming problems. Using these skills of inductive or deductive problem solving to understand and design computer language programs may facilitate
the development of design and help develop one’s aesthetic appreciation of “good” programs.

7. The two female problem-solving groups scored higher than the non problem-solving group on the problem-solving post-test. It might be that programming learning also affects development of problem-solving skills (Choi & Repman, 1993). There is a relationship between problem solving and programming which may have enabled female students to perform better on the problem-solving post-test.

Conclusions

This research studied the effect of first learning problem-solving skills to enable one to learn how to program in the BASIC language. The BASIC language programming midterm two and the BASIC language programming post-test provided the means to assess achievement in BASIC language program learning after learning problem-solving methods—induction or deduction. The data indicated that when female students study problem-solving methods (induction) and problem solving (deduction) they experience a significant increase in BASIC language programming achievement. Likewise, male students who learn problem-solving (induction) experience a significant increase in BASIC language programming achievement.
This study showed that female students who first receive problem-solving instruction in induction subsequently learn BASIC language programming significantly better than female students who first receive problem-solving instruction in deduction and subsequently learn BASIC language programming.

Further evidence supports that female students who first learn problem solving (deduction or induction) followed by receiving instruction in program design and understanding perform significantly better than female students who use a non problem-solving method (wordprocessing) prior to learning BASIC language programming in design and understanding. In addition, male students who first receive instruction in problem solving using induction perform significantly better than males who first learn non problem-solving instruction prior to learning BASIC language programming in design and understanding.

This study supports the finding that female students who receive instruction in problem solving using induction learn significantly better than female students who receive problem-solving instruction using deduction prior to learning BASIC language programming in understanding.

The finding of this study has implications for the learning suggested on Basic Theoretical Understanding of this study. This theory proposed that a mutual causation and interaction between problem solving and programming exists. This study provides partial support that learning a problem-solving method experience increases achievement in computer language programming. Many researchers (Au
& Leung, 1991; Reed et al. 1979, 1988; Saloman & Perkins, 1985) provided partial support that learning computer language programming may improve a learner's problem-solving abilities. Thus, when considering mutual causation both types of studies are combined, both "problem solving" and "computer programming" involve a common subset of cognitive behavior, memorizing and a schema or template. This supports the thesis that learning either problem-solving methods or programming provides a set of experiences which enhance the learning of the other.

Finally, this study has found evidence in support of the purpose of the study, that learning problem-solving methods (induction and deduction) subsequently increases achievement in BASIC language programming in design and understanding.

**Recommendations**

In this section, several useful recommendations of this study are made for BASIC language teaching. The purpose is to provide suggestions for teachers and students in learning BASIC language programming, and for further studies in programming teaching.

**Recommendation for teachers college students**

Learning computer programming should be widely used as educational activity. There were significant differences among students who learned problem-solving methods and non problem-solving methods which subsequently affected
learning BASIC language programming. Thus, it is strongly suggested that before learning BASIC language programming, students should increase their ability to employ problem-solving methods of deduction and induction (perhaps with an emphasis on inductive reasoning). By introducing problem-solving methods before introducing the construction of a program, teachers can better communicate to their students how to design and understand their programs.

Using problem-solving methods can help inexperienced teachers college students improve their effectiveness and establish confidence and experience in BASIC language program learning. It also enhances teachers college students' ability to know what problem-solving methods they should consider when working on BASIC language programs. Overall, the effective use of problem-solving methods will reduce misunderstandings and help students to acquire structure in employing problem-solving methods to increase their ability to learn BASIC programming.

The findings also have shown that a significant difference existed between induction and deduction methods on subsequent achievement in BASIC language program learning. Because novice students may lack experience in programming, when designing a program, induction should be considered as an important part of design and understanding. Using problem-solving methods and induction methods to develop one's own BASIC language program will help students not only to understand the program's results, but also help them to develop skill and ability to
design programs independently. Furthermore, it will enhance the quality of the program design.

**Recommendation for further study**

The recommendations for further study are based on the conclusions and findings of this study.

1. This study found significant differences between problem-solving methods and non problem-solving methods. Since there is limited research regarding the effects of deduction and induction problem-solving methods on subsequent achievement in the BASIC language, more detailed studies should be conducted using other problem-solving methods.

2. It was found that learning problem-solving methods subsequently increased proficiency in BASIC language programming. Future studies could focus on other programming languages such as LOGO, C, and PASCAL.

3. Since the study was limited by the number of subjects, particularly male subjects, further study with a larger sample should help clarify the effects for males. A study with equal numbers of males and females in the different treatments would permit examination of the interaction between gender and treatment.
REFERENCES


ACKNOWLEDGMENTS

During the years I pursued this final task of my doctoral research, so many people have contributed to make this research possible. I am especially thankful to the following people. First, I would like to thank Dr. William Miller, who advised me on this dissertation, and gave me valuable support and help. This dissertation was enhanced by his professional expertise and commitment. Many thanks to the other members of my committee: Dr. John Dugger who supported me throughout my doctoral program; Dr. Roger Smith who gave me many ideas about research; and Dr. Rex Thomas and Dr. John Boysen who offered expert advice and patience regarding the problem solving and programming ideas used in this dissertation. The participation of each committee member in my doctoral study helped to make it a rewarding experience.

There comes to mind many others who deserve thanks, including those who helped me develop and conduct my experimentation in the U.S. and Taiwan: the graduate students and teachers who modified the test questions and participated as instructors in the study. To my editor and friend, Pat Hahn, thank you for providing time, your computer, and assistance to complete this dissertation. To the many friends I made in Iowa State University and Ames, thank you for all the nice times we shared. In the past three years, we all helped one another to reach our final goal.
With the spirit and support from my family, I have had the courage to pursue my doctoral degree in the United States. I greatly appreciate their limitless support. Deep gratitude is expressed to my parents and father-in-law for their moral guidance and encouragement throughout my doctoral study. Finally, I am deeply thankful to my wife, Hsiu-chen Lin, and my two children, Ya-chun and Chen-wei, whose faith and belief in my abilities, tireless support, and permission to borrow three years from “our time” helped me complete my studies. I love you dearly, and I cherish your phone calls, letters and your visits. They kept us close together even though we were apart too much.
APPENDIX A: HUMAN SUBJECTS REVIEW
Information for Review of Research Involving Human Subjects
Iowa State University
(Please type and use the attached instructions for completing this form)

1. Title of Project: The effect of learning problem solving methods on learning in the BASIC language.

2. I agree to provide the proper surveillance of this project to insure that the rights and welfare of the human subjects are protected. I will report any adverse reactions to the committee. Additions to or changes in research procedures after the project has been approved will be submitted to the committee for review. I agree to request renewal of approval for any project continuing more than one year.

Hung, Yen-chih 02/10/95

3. Signatures of other investigators: Date: Relationship to Principal Investigator

William G. 2/25/95 Major Professor

4. Principal Investigator(s) (check all that apply)
- Faculty
- Staff
- Graduate Student
- Undergraduate Student

5. Project (check all that apply)
- Research
- Thesis or dissertation
- Class project
- Independent Study (490, 590, Honors project)

6. Number of subjects (complete all that apply)
- Adults, non-students
- ISU student
- Minors under 14
- Minors 14 - 17

7. Brief description of proposed research involving human subjects: (See instructions. Item 7. Use an additional page if needed.)

See Attachments

(Please do not send research, thesis, or dissertation proposals.)

8. Informed Consent:
- Signed informed consent will be obtained. (Attach a copy of your form.)
- Modified informed consent will be obtained. (See instructions. Item 8.)
- Not applicable to this project.
Checklist for Attachments and Time Schedule

The following are attached (please check):

12. ☐ Letter or written statement to subjects indicating clearly:
   a) purpose of the research
   b) the use of any identifier codes (names, #'s), how they will be used, and when they will be
      removed (see Item 17)
   c) an estimate of time needed for participation in the research and the place
   d) if applicable, location of the research activity
   e) how you will ensure confidentiality
   f) in a longitudinal study, note when and how you will contact subjects later
   g) participation is voluntary; nonparticipation will not affect evaluations of the subject

13. ☐ Consent form (if applicable)

14. ☐ Letter of approval for research from cooperating organizations or institutions (if applicable)

15. ☐ Data-gathering instruments

16. Anticipated dates for contact with subjects:
   
<table>
<thead>
<tr>
<th>First Contact</th>
<th>Last Contact</th>
</tr>
</thead>
<tbody>
<tr>
<td>02/25/95</td>
<td>06/30/95</td>
</tr>
</tbody>
</table>

   Month / Day / Year       Month / Day / Year

17. If applicable: anticipated date that identifiers will be removed from completed survey instruments and/or audio or visual tapes will be erased:

   12/31/95
   Month / Day / Year

18. Signature of Departmental Executive Officer       Date       Department or Administrative Unit

19. Decision of the University Human Subjects Review Committee:

   ☑ Project Approved       ☐ Project Not Approved       ☐ No Action Required

   Patricia M. Keith
   Name of Committee Chairperson       Date       Signature of Committee Chairperson
Dear Dr. William G. Miller

Yen-chu Hung is a member of the faculty in the Department of Math & Science of National Chia-yi Teachers College. His study titled "The Effect of Learning Problem Solving Methods on Learning to Program in the Basic Language" has been permitted to conduct at this college. The guideline of human subject established by Research Committee of Iowa State University will be respected at this study.

Sincerely yours,

Kuo-Hsiung Chen
President
National Chia-yi Teachers College

Feb. 17, 1995
APPENDIX B: PRETEST MATERIAL
Basic Pretest

The pretest is being given to estimate how much you may already know about using a computer. Your score will not affect your course grade. It will help your instructor plan the course lessons.

Student identification number:____________________
Name:____________________

Directions:
A. There are several types of questions in this instrument. Some questions require that you record the letter of the best choice in the blank provided. Other questions ask you to write a short answer in the space provided. Please do your best to answer all questions.
B. There are 20 questions and you have 50 minutes to complete the test. Go ahead.

1. Which of the following is not considered computer hardware?
   a. CPU  b. Basic Compiler  c. Disk  d. PROM  e. Modem

2. Which of the following is a function of RAM (random-access memory)?
   a. to hold a program  b. to provide system power
   c. to speed up the computer  d. to convert keystrokes to binary numbers
   e. to print character on the printer

3. What is the purpose of an Interface card?
   a. to connect to a power supply  b. to connect other devices to the CPU and memory
   c. to transform BASIC statement to machine language  d. to speed up numerical processing
   e. to execute a program

4. DOS is
   a. Spreadsheet software  b. Computer Aided Design software
   c. Statistical Analysis software  d. Data Organization software
   e. Operating System software

5. What does the term "path" mean when we are using a computer?
   a. Getting directions from an expert
   b. The flow of electrons from CPU to RAM  c. The description of where data is located
   d. The organization of a computer program  e. The route that telecommunications takes to our computer.
6. The difference between "cold starting" and "warm starting" a computer is?
   a. No difference
   b. It depend on the weather cold or warm
   c. A different method in the shutdown computer
   d. A difference by the expert or novice using
   e. Use Keyboard restart and computer power i/o shutdown or start

7. What is the difference between the RAM (random-access memory) and
   ROM (read-only memory)?
   a. The manufacturer supplying it
   b. The speed with which you can read data level
   c. The physical size of the circuits
   d. The ability to store frequently changing data
   e. The degree of difficulty to install in the computer

8. Which of the following is a "source language" file?
   a. mine.exe
   b. mine.com
   c. mine.pas
   d. mine.pcx
   e. mine.obj

9. Compared to other teachers, I think I am
   a. much more comfortable with computers
   b. slightly above the average in comfort in using computers
   c. About average in comfort or anxiety in using computers
   d. Slightly more anxious when computers than most teachers
   e. A lot more anxious about using computers.

10. What output is displayed by the following program?

    10  A = 8
    20  B = 4
    30  C = (A + B) /2
    40  PRINT "The average of ", A; " and "; B; "is" ; C
    50  END

    a. The average of A and B is C
    b. The average of 8
       and 4
       is 6
    c. 6
    d. The average of 8 and 4 is 6
    e. The average of A8 and B4 is C6
11. What output is displayed by the following program?

```
10 FOR I = 1 TO 5
20 A = A + I
30 NEXT I
40 PRINT A
50 END
```

a. 5  b. 15  c. 0  d. 20  e. no any output

12. What output is displayed by the following program?

```
10 X = 2
20 Y = X^2
30 X = X + 1
40 IF X < 3 THEN GOTO 20
50 PRINT Y
```

a. 8  b. 3  c. 9  d. 4  e. 16

Part II Short answers. Record a short answer to each question below the question

13. Which Wordprocessing programs have you used?

14. Which spreadsheet software have you used?

15. Construct a flow chart for finding the largest of three numbers.

16. What is difference between the "interpreter" and "Compiler" in the computer language?

17. What is difference between "Top-down design" and "bottom up design" in writing programs?

18. What is structured programming?

19. Have you written your own computer programs? If Yes, in what language (BASIC, C, PASCAL, FORTRAN, LOGO,..etc)

20. Have you had other courses that required use of computers? If Yes, please describe it.
Problem-solving Pretest

There are several types of questions in this instrument. Some questions require that you record the letter of the best choice in the blank provided. Other questions ask you to write a short answer in the space provided. Please do your best to answer all questions.

Student identification number: ________________________
Name: ________________________

Direction:
There are 20 questions and you have 50 minutes to complete the test. Go ahead.

1. All C are B
   No B are A
Which of the following conclusions can be deduced from the two premises above?
I. No C are A
II. All B are A
III. All C are A
   a. I only       b. II only
   c. I and II only   d. III only
   e. II and III only

2. The county of Chiayi contains six towns.
   F is the westernmost city and south of E
   D and E are south of C
   A and D west and south of E and south of F
   B is east of C and south of D
   C is east of E
   The Southernmost town in Chiayi is
   a. A       b. B
   c. C       d. D
   e. E

3. Question 3 refer to the following statements
   1. Animals can outrun any animals that they eat
   2. Carnivores eat other animals
   3. Outrunning is transitive i.e., If x can outrun y and y can outrun z, then x can outrun z.
   4. Lions eat zebras
   5. Zebras can outrun dogs
   6. Dogs are carnivores
   Which conclusion is true.
   a. Zebras outrun Lion
b. Lion outrun Zebras
c. Dog outrun Lions
d. Zebras are carnivores.
e. One cannot draw a conclusion

4. All dogs are animals. Some predators are dogs. Therefore, some predators are animals. The syllogism above is
   a. stained
   b. illogical
   c. internally inconsistent
   d. valid
   e. invalid

5. Question 5 refer to the following statements.
   A, B and C are three chemical elements
   If C reacts with C, the result is B
   If A reacts with C, the result is C
   If B reacts with any element, the result is always B.
   If A reacts with A, the result is A.
   The order of the reaction makes no difference.
   Which of the following must be true?
   I If A reacts with any other element, the result is that element
   II If B reacts with B, the result is C
   III If C reacts with any other elements, the result is never C
   a. I only
   b. I and II only
   c. II only
   d. III only
   e. II and III only

6. Questions 6 refer to the following information.
   A salesman must visit four cites.
   The cites he may choose from are Taipei, Taoyuan, Taichung, Chiayi, Tainan, and Kaoshiung.
   The salesman must visit either Chiayi or Tainan, but not both.
   For some reason, the salesman cannot visit Taichung and Kaoshiung together.
   If a salesman visits Chiayi, which other cities must he visit?
   I Kaoshiung
   II Taoyuan
   III Taichung
   a. I only
   b. II only
   c. III only
   d. I and II only
   e. II and III only

7. Questions 7 refer to the following information.
   There are five students, A, B, C, D, and E. All take computer training. They are five different teaching activities, 1, 2, 3, 4, and 5. All students will engage in four or more activities each day and must spend at least 20 minutes on any activity once they begin it. No more than three students
can be at any one activity at the same time. A and B cannot work together. C and D must always work together. The student are at school from 9 a.m. to 12 noon and take a 30 minute rest from 10:15 to 10:45 daily. Which of the following groupings conforms to the conditions for organizing the student into activities during the period from 11:00 - 11:30
a. B, C and E - Activity 4; A and D - Activity 2
b. A, C and D - Activity 2, B and E - Activity 1
c. A, C, D and E - Activity 4; B - Activity 3
d. A, B, and C - Activity 5; D and E - Activity 4
e. A and C - Activity 2; B, D and E - Activity 3

8. Question 8 refer to the following statements
A teacher Lineup contained four students(A, B, C, D), one of whom is a gifted student.
The lineup is graduated by height. The tallest student on the left and shortest on the right
There are two students between A and B
C is the to left of D
The gifted student is third from the left
B is to the right of the gifted student
Who is the gifted student?
a. A  b. B  c. C  d. D  e. cannot be determined

9. There are 8 players in a one-on-one basketball tournament. Each player must play each other player one game. How many games will be played in the tournament?
a. 56  b. 8  c. 28  d. 24  e. 4

10. John fell asleep after dinner one night. When he awoke he realized that he would have to rush in order to make the last call at the Soul-Ace-Hotel. He grabbed a T-shirt and put it on inside-out, with his left arm in the right sleeve and his right arm in the left sleeve. Where is the label?
a. outside back  b. inside back  c. outside front  d. inside front

11. Student A is shorter than student B
Student C and student B are the same height.
Student D is taller than student C
Student E is taller than student A
If the above statements are true, which of the following must also be true?
a. Student E is taller than student C  
b. Student E is taller than student D  
c. Student B is shorter than student E  
d. Student C is shorter than student A  
e. Student D is taller than student A  

12 All A are B. Some C are A. The invalid condition to these two propositions is  
a. Some B are A  
b. Some C are B  
c. Some A are C  
d. All C are B  
e. Some C aren't B  

13. Question 13 refers to the following statements.  
I. All wheeled conveyances which travel on the highway have more than two wheels  
II. Bicycles do not have more than two wheels  
III Whenever Mary drives her car on the highway, it is sunny.  
IV It is sunny  
If the above statements are all true, which of the following statements must also be true?  
a. Bicycles do not travel on the highway.  
b. Bicycles travel on the highway only if it is sunny  
c. If Mary's car is not more than two wheels, then it is not sunny  
d. Mary are now driving her car on the highway  
e. Mary's car is not more than two wheels.  

Part II Short answers. Record a short answer to each question below record your answer below the question  

14 There are seven coins which look identical. One of the seven coins weighs slightly less than the other six. Using a balance scale, how could you determine which is the light coin in just two weightings?  

15. The problem is the “Tower of Hanoi” problem, whose initial setup is shown in below. Three pegs A, B, and C, exist. Five disks of differing diameters are placed on peg A so that a larger disk is always below a smaller disk. The object is to move the five disk to peg C using peg B as auxiliary. Only the top disk on any peg may be moved to any other peg, and a large disk may never rest on a smaller one. What is your procedure?
16. Consider the eight queens problem: given an 8 x 8 chess board that is initially empty, and 8 queens, the task is to add one piece at a time subject to the constraint that no two pieces can occupy the same row, the same column, or the same diagonal, so as to end up with all eight queens on the board. What is your procedure?

17. A worm is at the bottom of forty meter hole. It can crawl upwards at the rate of four meter in one day, but at night, it slips back three meters. At this rate, how long will it take the worm to crawl out of the hole?

18. Ben can never tell a lie. George can never tell the truth. One of them said, "The other one said he is George." Which one said that?

19. If two hours ago, it was as long after one o'clock in the afternoon as it was before one o'clock in the morning, what time would it be now?

20. Two mothers and two daughters were fishing. They managed to catch one big fish, one small fish, and one fat fish. Since only three fish were caught, how is it possible that they each took home a fish?
Basic programming post-test

There are several types of questions in this instrument. Some questions require that you record the letter of the best choice in the blank provided. Other questions ask you to write a short answer in the space provided. Please do your best to answer all questions.

Student identification number: ________________________
Name: ________________________

Direction:
There are 20 questions and you have 50 minutes to complete the test, Go ahead.

1. Rewrite the following group of statements using a FOR loop
   100 k = 5
   110 IF k > 15 THEN 150
   120 x = x + k
   130 k = k + 3
   140 GOTO 110
   150 < next statement>

2. What will be the value of variable f after the statements below are executed?
   10   f = 0
   20 FOR i = 1 to 5
   30 IF g(i) < 25 then f = 1
   40 NEXT i
   Given the array g as shown g(1) = 12, g(2) = 25, g(3) = 40, g(4) = 0, g(5) = 300

3. Write a program for an algorithm to compute the factorial, N!, of a single arbitrary integer N. (N! = N x (N - 1) x (N - 2) x .... x 2 x 1).

4. Write a BASIC program to compute and print a table showing the conversion from degrees Celsius to degree Fahrenheit for temperatures ranging from 0°C to 100°C in step of 10°C.
   The formula is Fahrenheit = 1.8 x Celsius + 32
<table>
<thead>
<tr>
<th>Celsius</th>
<th>Fahrenheit</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>32</td>
</tr>
<tr>
<td>10</td>
<td>50</td>
</tr>
<tr>
<td>20</td>
<td>68</td>
</tr>
<tr>
<td>30</td>
<td>86</td>
</tr>
</tbody>
</table>
5. List the values printed as the loops below are executed

```
10  n = 77
20  d = 3
30  DO  WHILE  d <= SQRT(n)
   40  PRINT  d
   50  d = d + 2
   60  LOOP
```

6. What statement in 20, Could get the following answer?

```
10  FOR  i = 1 to 2
20      1  1
30  PRINT  i ; j
   31  1  3
40  NEXT  i
   41  2  1
50  NEXT  j
   51  2  2
60  END
   61  2  3
```

7. How many lines will be displayed when the following group of statements is executed?

```
10  j = 20
20  IF  j <= 3 THEN 60
30      PRINT  j
40      j = j - 4
50    GOTO 20
60 < next statement>
```

8. Assume that the value of A is 2, B is 3, and C is 4. What output is displayed by the following group of statements?

```
10  PRINT  A ; B ,
20  PRINT  C ;
30  PRINT  "data"
```

9. Assume that the value of X$ is xyz, Y$ is 123, and Z$ is abc. What is the value of the following string expressions?

```
10  A$ = Z$ + X$
20  B$ = LEFT$(X$,1) + MID$(Y$,2,1)+ RIGHT$(Z$,1)
30  C$ = A$ + "sdf" + B$
40  PRINT  C$
```

10. Write a program that inputs a string, for example “abcdef”, and then outputs the inverse, for example “fedcba”.

   input a string : “abcdef”
   the output is “fedcba”
11 What is wrong with the following program?
10 INPUT x
20 GOSUB 40
30 PRINT y
40 y = 2 * x
50 RETURN
60 END

12 What output is displayed by the following program?
10 DEF FNA(b) = a + b
20 a = 5
30 b = 10
40 a = FNA(a)
50 b = FNA(b)
60 PRINT a, b

13 Convert the following English descriptions of algorithms to flow diagrams and BASIC statements.
   a.) If the remainder(r) is equal to zero, then print n
   b.) If the product(p) is equal to n, then print the contents of the variable d and read a new value into n

14. What statement in 50, Could get the following answer?
10 FOR I = 1 TO 5  #
20 FOR j = 1 TO I  ##
30 PRINT "#";  ###
40 NEXT j  ####
50  ######
60 NEXT i

15 What output is generated from the following programs?
30 r = 7 : h = 49
40 GOSUB 500
50 PRINT "cross pay is ", g
60 END
500 g = r * SQR(h)
510 RETURN

16 Write a program to compute the future value of money deposited at a bank. The formula is: Future value = Amount (1 + Interest rate) 
You need write input statement to input the Amount, interest rate, year
17. What output is displayed by the following program?

```
10 FOR I = 1 TO 3
20 FOR J = 1 TO 2
30 READ s(i, j)
40 NEXT j
50 NEXT I
60 REM - display array data
70 FOR I = 1 TO 3
80 FOR J = 1 TO 2
90 PRINT s(i, j),
100 NEXT C
110 PRINT
120 NEXT I
130 DATA 91, 78, 85, 95, 96, 90
140 END
```

18. Code a group of statements that computes the length of a room in yards and feet, given the length in feet. For example, if the length is 17 feet, the result should be 5 yards and 2 feet.

19. Write a BASIC program to evaluate the following score?

90 - 100 A  
80 - 90 B  
70 - 80 C  
60 - 70 D  
0 - 60 F

20. Write a program to find which student(s) score > 95, and print that student's Id number (1-15) and his or her score. The data are presented below:

<table>
<thead>
<tr>
<th>Id number</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>student</td>
<td>s(1) = 87 , s(2) = 67 , s(3) = 99 , s(4) = 89 , s(5) = 100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>score</td>
<td>s(6) = 78 , s(7) = 59 , s(8) = 95 , s(9) = 93 , s(10) = 98</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>s(11) = 82 , s(12) = 95 , s(13) = 85 , s(14) = 92 , s(15) = 46</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Problem-solving Posttest

There are several types of questions in this instrument. Some questions require that you record the letter of the best choice in the blank provided. Other questions ask you to write a short answer in the space provided. Please do your best to answer all questions.

Student identification number: __________________________.
Name: __________________________.

Directions:
There are 20 questions and you have 50 minutes to complete the test. Go ahead.

1. If you have a car which was manufactured in Japan after 1985, it has excellent safety features. The statement above can be deduced logically from which of the following statements?
   a. Excellent safety features were not developed in the United States until 1985.
   b. Only if a car was made after 1985 could it have excellent safety features.
   d. Some cars made in Japan before 1985 had excellent safety features.
   e. All cars manufactured in Japan after 1985 have excellent safety features.

2. During February of this year, it always rained more than three inches per day in Taiwan where the temperature was 17°C or higher. Temperatures in February ranged from 10°C to 25°C.
   Assuming that the statements above are true, which of the following CANNOT be an accurate report of the temperature and rainfall on a February day this year in Taiwan?
   a. 10°C, 4"     b. 15°C, 0"
   c. 17°C, 3.5"   d. 18°C, 3"
   e. 20°C, 5"

3. Two women A and B and two men C and D are teachers. One is teaching Math, one teaching History, one teaching Science and one teaching English. They are seated around a square table with one person on each side.
   1) C is across from the Math teacher
   2) D is not across from the teaching History teacher
   3) The Science teacher is on the A’s left.
   4) B is the English teacher.
   5) The History teacher and English teacher are married to each other.
   6) The English teacher is not on B’s left
7) The English teacher is across from the teaching Science. Which of the following must be false?
I. C is the Math teacher
II. The History and English teachers are women
III. The Math teacher is across from the History teacher
a. I only b. II only
c. III only d. I and II only
e. II and III only.

4. Mr. K is hiring five person to do program and art designing on a new project. He must have a minimum of two programmers. Nine persons have applied for the job: A, B and C are programmers, while D, E, F, G, H, and J are art designer.
1) Mr. K is unwilling to hire G and H to work together because they argue all the time.
2) E and F are buddies and will only work together.
3) C won't work with D because of their failure in a limited partnership effort.
If A, B and C are hired, the team of art designers can consist of
a. only E and F
b. E and F or G and H
c. G and J or H and J
d. E and F, or G and J, or H and J
e. D, E and J

5. "All actions have consequences. Given this fact, we may wish to play it safe by never doing anything."
The speaker implies that
a. we may prefer to live safely
b. all acts have consequences
c. consequentiality is not safe.
d. doing nothing is not an act and keeps us safe from consequences
e. not doing anything is not an act

6. A is older than B and taller than C
D is younger than E, older than C, and shorter than F
G is older than H, younger than C, shorter than H, and taller than F.
H is older than A and shorter than C
Which of the following is the youngest?
a. A b. B
c. C d. D
e. E

7. Person A said "All Chinese enjoy rice."
Person B said “I must disagree. I have known some Chinese who loved noodles.”
Person B response shows that he has interpreted Person A’s remark to mean that
a. Chinese do not like noodles
b. Only Chinese eat noodles
c. Most people cannot appreciate good rice
d. Only Chinese enjoy rice
e. Chinese enjoy only rice

**8.** "Careers and job opportunities are increasing in computer technology. In the next five years careers in the computer industry will exceed the opportunities in any other single field of work.” If the preceding statement is true, which of the following implications is the most reliable for young people making a choice about their future occupation and career?

a. More people will be working in the computer industry than all other fields of work combined.
b. Only those who prepare for work in the computer technology can expect a reasonable opportunity to find productive employment.
c. Those who enjoy high status such as professionals in medicine and law and corporation presidents will be replaced in status by computer expert.
d. Those who prepare for employment in the computer industry will most likely find productive employment.
e. Those who are unskilled or lacking knowledge that is useful in the computer industry will be destined to the lower paying jobs in the economy.

**9.** Person A buy 100 acres of land in an isolated area as an investment. The land was cheap and he thought the land would appreciate because people would seek solitude away from the city area. Which of the following events is Person A assuming will not happen?

a. The taxes and other costs in holding the property will become prohibitive for him.
b. Access to his property will not be unreasonable.
c. The property will not become valuable before he is too old or deceased.
d. The property will not be damaged by natural causes(flood, fire, erosion, etc).
e. The zoning ordinances will not prevent development into a profitable enterprise or sale.
10. Five managers of an international company hold a meeting in Taipei, Taiwan.
   1) Mr. A converses in Chinese and German
   2) Mr. B converses in Chinese and English
   3) Mr. C converses in English and German
   4) Mr. D converses in Japanese and Chinese
   5) Mr. E converses in German and Japanese
   Which of the following can act as interpreter when Mr. C and Mr. D wish to confer?
   a. Mr. A  
   b. Mr. B  
   c. Mr. E  
   d. Mr. A or Mr. B  
   e. Any of the other three

11. Five children in a family of six children have dimples. Three children in the family are girls. Four children in the family have black eyes. Which of the following must be true?
   I All of the girls have dimples
   II At least one girl has black eyes.
   a. I only  
   b. II only  
   c. Either I or II, but not both  
   d. Both I and II  
   e. Neither I nor II

12. Three cars are left in the Chiayi-i: a 1978 Ford, a bright red Toyota, and a blue Lincoln. The three drivers are A, B and C. Who is driving each car?
   I C drove into the Lincoln.
   II C said "I'm going to that Toyota!"
   III A saw the Toyota coming at him from the side.
   IV B saw the Ford hit the Lincoln
   a. B the driver of the Lincoln  
   b. A the driver of the Ford  
   c. C the driver of the Toyota  
   d. C the driver of the Ford  
   e. A the driver of the Toyota

13. Eight students A, B, C, D, E, F, G and H play audition for the same role, one after the other. Three of them B, F, and H sing. Four of eight A, C, D, and G dance. Student who have the same talent, singing or dancing, cannot audition one after another. Which of the following is a possible order of auditions?
   a. B C F D A E G H  
   b. A B C D E F G H  
   c. F A B C G D E H  
   d. C B A E F H D G  
   e. A B C E D F G H
Part II Short answers. Record a short answer to each question below the question

14 Mr. and Mrs. A have five children. Half of them are boys. How is this possible?

15 Professor B and his sister C were passing the "ABC Hotel" when the professor said. "My nephew is in there and I would like to stop and say hello." "Since I don't have a nephew, I'll continue on and see you later." replied C. What relation is C to the mysterious nephew?

16. There is a train one kilometer long, traveling at a rate of one kilometer per minute through a tunnel which is one kilometer long. How long will it take the train to pass completely through the tunnel?

17. Mr. and Mrs. P have six daughters and each daughter has one brother. How many people are in the P family?

18. When the day after tomorrow is yesterday, today will be as far from Wednesday as today was from Wednesday when the day before yesterday was tomorrow. What is the day after this day?

19. A married a widow. At that time they each had children of their own. Thirteen years later the A kids got into a heated argument. Mrs. A exclaimed to Mr. A, "Your children and my children are fighting with our children!". Each parent is directly related to nine of the 12 children in the A family. How many children were born after Mr. and Mrs. A were married?

20 In a meeting of 50 scientists and poets, 35 are scientists, 30 have short hair, 25 are scientist short hair, how many long-haired poets are there? (you may assume that no scientist is a poet.)
APPENDIX D: INSTRUCTIONAL MATERIAL
Problem-solving instruction (1)
Week one (two hours)
For deduction and induction group

OBJECTIVE:
1. To pretest the problem-solving ability and BASIC language ability.
2. How to learn Problem-solving skills (1).

THEORY:
(1) Understand the problem
   a. Do you understand all the words?
   b. Can you restate the problem in your own words?
   c. Do you know what is given?
   d. Do you know what the goal is?
   e. Is there enough information for solve the problem?
   f. Is there extraneous information?
   g. Is this problem similar to any other problem you have solved?

Example:
Four students (A, B, C, D) are lineup and assigned by a teacher, One of whom is a gifted student. The lineup is according to by height, with the tallest student on the left and shortest on the right. There are two students between A and B, and C is the left of D. The Gifted students is the third from the left. B is on the right of the gifted student. Who is the gifted student?
a. A  b. B
c. C  d. D
e. cannot be determined

(2) Devise a Plan
   Can one of the following strategies (heuristics) be used? (Say three Basic strategies)
   a. Look for a pattern
   b. Solve a simpler problem
   c. Draw a picture

Example:
How to get any two point distance from a cube? (include a diagonal line suppose we know the cube width, height, long?)

(3) Carry out the Plan
a. Implement the strategy or strategies that you have chosen until the problem is solved or until a new course of action is suggested.

b. Give yourself a reasonable amount of time in which to solve the problem. If you are not successful, seek hints from others or left the problem aside for a while.

c. Don't be afraid of starting over. Often, a fresh start and a new strategy will lead to success.

(4) Look Back

a. Is your solution correct? Does your answer satisfy the statement of the problem.

b. Can you find an easier solution?

c. Can you recognizing how you can extend your solution to a more general case.

PROCEDURE:

1. Pretest the student problem-solving and BASIC language program ability. (60 minutes)

2. Introduce problem solving theory (30 minutes).

3. Indicate that we will discuss the methods of problem solving (5 minutes)

4. Discuss examples of problem solving (25 minutes).
Problem-solving instruction(2)
Week two (one hour)
For deduction and induction group

OBJECTIVE:
1. How to learn Problem-solving skills(2).

THEORY:

(1) Problem-solving strategy: Draw a Diagram

Example:
A survey was taken of 150 college freshmen. Forty of them major in mathematics, 30 of them major in English, 20 major in science. 7 had a double major of mathematics and English, and none had a double(triple) major with science. How many students had majors other than mathematics, English, or science?

Discussion:(let student group discuss)
A venn diagram with three circles, show below, is useful in this problem.

There are 150 students within the rectangle, 40 in the mathematics circle, 30 in the English circle, 20 in the science circle, and 7 in the intersection of the mathematics and English circles but outside the science circle. There are 83 students (33 + 7 + 23 + 20) accounted. There must be 67 students (150 -83) outside the three circles. Those 67 students were the ones who did not major in mathematics, English, or science.

(2) Problem-solving strategy: (Use Direct Reasoning)

Example:
In a group of nine coins, eight weigh the same and the ninth is heavier. Assume that the coins are identical in appearance. Using a pan balance, what is the smallest number of weightings needed to identify the heavy coin?

Discussion:(let student group discuss)
Two weightings are sufficient. Separate the coins into three groups of three coins each.

\[
\begin{array}{ccc}
O & O & O \\
A & B & C
\end{array}
\]
Weigh group A against group B. If they balance, we can deduce that the heavy coin is in group C. In this case, select two coins from group C. Weigh one against the other. If they balance, the remaining coin in group C is heavy. If they do not balance, the heavier coin tips the scales. If the coins in group A do not balance the coins in group B, one group is heavier and must contain the heavy coin. Assume that is in group A. Then choose two coins from group A and weigh one against the other. Use the same reasoning as in the preceding paragraph with coins from group C. In any case, Two weighing are needed.

(3) Problem-solving strategy: (Working Backwards)

Example:

How can you bring up from the river exactly six quarts of water when you have only two containers, a four quart pail and a nine quart pail, to measure with?

Discussion: (Let student class discuss)

Let us visualize clearly the given tools we have to work with, the two containers. (What is given?) We imagine two cylindrical contains having equal bases whose altitudes are as 9 to 4. There is no scale. How get the solution?

We do not know yet how to measure exactly 6 quarts? If the large get exactly 6 quarts in it and the smaller container empty. If We can fill the large container to full capacity, that is, to 9 quarts. But then we should be able to pour out exactly three quarts. To do that We must have just one quart in the smaller container! That's the idea.

The situation of this, We fill the large container to full capacity, and pour it four quarts into smaller container and then into the river, twice in succession. We can get the one quarts in the container. We came eventually upon something already known and following the method of analysis, working backwards.

PROCEDURE:

1. Discuss examples of problem solving question. (10 minutes)
2. Introduce problem solving theory (20 minutes).
3. Indicate that we will discuss the methods of problem solving (5 minutes)
4. Group discuss the problem-solving methods (20 minutes).
5. Write a short conclusion (5 minutes).
OBJECTIVE:

1. How to learn Problem-solving skills(3).

THEORY:

(1) Problem-solving strategy (Setting up equations)

Setting up equations is like translation from one language into another.

a. A condition, to set up equations means and to express in mathematical symbols (computer language) that is stated in words; It is translation from ordinary language into the language of mathematical formulas (program design).

b. We must be familiar with the forms of mathematical expression (computer language expression).

(2) Problem-solving strategy (Analogy)

Analogy is a sort of similarity. Similar objects agree with each other in some respect, analogous objects agree in certain relations of their receptive parts. Examples:

A rectangular parallelogram is analogous to a rectangular parallelepiped. The relations between the sides of the parallelogram are similar to those between the faces of the parallelepiped.

(3) Problem-solving strategy. (Decomposing and recombining)

Decomposing and recombining are important operations of the mind. You examine an object that attracts your interest or challenges your curiosity; a house is intend to lease, an important but cryptic telegram, any object whose purpose and origin puzzle you, or any problem you intend to solve. You have an impression of the object as a whole but this impression, possibly, is not definite enough. A detail strikes you, and you are focused your attention upon it. then, you concentrate upon another detail; then, again, upon another. Various combinations of details may present themselves and after a while you consider the object as a whole again but you see it now differently. You decompose the hole into its parts, and you recombine the parts into a more or less different whole.
(4) Problem-solving strategy (Heuristic)

Heuristics are strategies to enable those who, after having studied ordinary elements of concept, may desire to acquire the ability to understand and advance their ability to solve mathematical problems (program design). Heuristics are used to teach the procedures of analysis and synthesis. In analysis, one starts from what is required, or taken for granted, and one then draws conclusions, and later develops problem-solving methods to deal with future events, until a point is reached where one can use new knowledge as the starting point in synthesis.

In synthesis, reversing the process, one starts from the point which was reached following the analysis, that is from what is already known or admittedly true. One derives from the preceding analysis, and continues to make derivations until, retracing steps, success is achieved in understanding what is required. This procedure is called synthesis, or constructive solution or progressive reasoning.

PROCEDURE:

1. Discuss examples of problem-solving question (10 minutes).
2. Introduce problem-solving theory (20 minutes).
3. Indicate that we will discuss the methods of problem-solving (5 minutes).
4. Group discuss the problem-solving methods (20 minutes).
5. Write a short conclusion (5 minutes).
Problem-solving instruction(4)
Week four (one hour)
For deduction group

OBJECTIVE:
1. To learn the problem-solving method - deduction.

THEORY:
When studying useful forms of induction, the thinker is likely to formulate a symmetrical form of analysis which will reduce the problems that exist to their simplest form. For example, a deductive inference could be stated as: A personal computer has a 100 MHz processor. Therefore, one can deduce that this computer is faster than earlier generations of the technology. Therefore, this requires lateral knowledge of the design of computer technology and electronics to recognize the clock speed of the computer. Without this related peripheral knowledge, one lacks the tools to deduce. Therefore, deduction requires a broad based experienced researcher or the technology available through a computerized environment. This deduction strategy can be developed in several ways. (Let students discuss within the group, each group having three to five students. The students review all the problem-solving methods and say what is(are) the deduction methods)

PROCEDURE:
1. Introduce problem-solving method - deduction (15 minutes).
2. Discuss examples of deduction (15 minutes).
3. Indicate that we will discuss the methods of deduction (5 minutes)
4. Group discuss the problem-solving method - deduction (15 minutes).
5. Group speak about deduction method (5 minutes)
6. Write a short conclusion (5 minutes).
Problem-solving instruction (4)
Week four (one hour)
For induction group

OBJECTIVE:

1. To learn Problem-solving method - induction.

THEORY:

With a type of problem solving called the inductive reasoning process, many new facts are based on and related to what is already known, and at last a new whole is formed. Creative solutions are demanded of inductive thinkers. Such thinkers will have to organize, retrieve, and use a excess of information to solve their problems. Induction uses experimental reasoning to arrive at the whole from the particulars. The initial formulation of constructive induction uses domain knowledge to develop new concept or attributes beyond those supplied in the input (Michalski, 1983). Thus, induction is a basic inference strategy used in synthesized learning.

PROCEDURE:

1. Introduce problem-solving method - induction (15 minutes).
3. Indicate that we will discuss the methods of induction (5 minutes)
4. Group discuss the problem-solving method - induction (15 minutes).
5. Group speak about induction method (5 minutes)
6. Write a short conclusion (5 minutes).
APPENDIX E. COVER LETTER AND SURVEY INSTRUMENT
Dear Student Evaluator:

I am conducting a research study in the area of evaluation of learning problem-solving methods on learning to program in the BASIC language to complete the dissertation requirement in my program of study at Iowa State University. Your cooperation is being sought to gather data about the comparison of problem-solving and non-problem-solving methods for the evaluation on learning to program in the BASIC language.

Please complete the enclosed evaluation form. It will take approximately 50 minutes to complete. Your participation is voluntary and any information that is provided will be kept strictly confidential. All data will be analyzed and reported as group data only. Your experience and knowledge in learning to program in the BASIC language is beneficial to the success of this research.

I greatly appreciate your assistance in this study. If you have any questions, please feel free to discuss them with me.

Sincerely

Hung, Yen-chu

William G. Miller
Professor&Major Advisor
APPENDIX F. BASIC LANGUAGE PROGRAMMING MODULES
The BASIC language programming modules

(1) An initial BASIC tutorial on the operations of the microcomputer system
   1. Introduction
   2. Chinese version QuickBASIC introduction

(2) A guide to starting BASIC on the PC
   1. Use Chinese operation system
   2. Start in QuickBASIC
   3. End on QuickBASIC

(3) A guide on Writing a Program
   1. A description of line numbering, in the BASIC program.
   2. QuickBASIC work environments introduction
   3. New programming development
   4. Run the programming

(4) QuickBASIC work function
   1. Make a exe programming
   2. Save a new programming
   3. Load a programming

(5) Watch the monitor
   1. The PRINT statement
   2. String and things you can see

(6) Memory and mathematics
   1. Values and variables
   2. Storage space for numbers
   3. Calculation by Computer

(7) What is program?
   1. Introduction the program
   2. Comment and flow chart

(8) Questions and answer
   1. If - then statement
   2. What comes next?
   3. And then he said....
   4. More decision
(9) Out of order
   1. GOTO statement
   2. For - next loops
   3. Do - while loops
   4. Program jump - from line to line
   5. Again... and Again ..... and Again

(10) Timers and counters
   1. More about loops
   2. Nested loops - a loop inside a loop inside

(11) Coming Input - going Output
   1. Data from the keyboard
   2. More input about input
   3. Getting data in and out

(12) Where the information goes
   1. Values into variables
   2. More information about data
   3. Data on demand

(13) Charting the course
   1. Flowcharts and branching
   2. Pathways through programs
   3. Go with flow

(14) Programming top to bottom
   1. Keep the user informed
   2. More friends with a computer
   3. Best fact forward
   4. Read it again

(15) For use and reuse
   1. Subroutines
   2. reusable routines
   3. Errors to avoid

(16) Computer delights
   1. BASIC functions
   2. Move it over
   3. Pick a number
(17) Summing it up
   1. Accumulating values
   2. Total to average
   3. Count on your computer

(18) Keeping it together
   1. Subscript variables
   2. Lot of lists
   3. Playing with arrays
   4. Sets of string
   5. Bits of string

(19) Computer music and graph
   1. Play a sound
   2. Play a song
   3. Draw a line, circle
   4. Paint, pset
   5. Draw a simple graph

(20) File manager
   1. Sequence files
   2. Randomly files
   3. Simple database use file control