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.

U·M·I

University Microfilms International
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313/761-4700 800/521-0600

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

Order Number 9309130

The effect of closed laboratory activities on the comprehension of five concepts and the perception of effectiveness of the course in a second semester computer science course

Burton, Debra Lynn, Ph.D.

The University of Texas at Austin, 1992

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

THE EFFECT OF CLOSED LABORATORY ACTIVITIES ON THE COMPREHENSION OF FIVE CONCEPTS AND THE PERCEPTION OF EFFECTIVENESS OF THE COURSE IN A SECOND SEMESTER COMPUTER SCIENCE COURSE

by

DEBRA LYNN BURTON, B.S., M.S.

DISSERTATION

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

December 1992

THE EFFECT OF CLOSED LABORATORY ACTIVITIES ON THE COMPREHENSION OF FIVE CONCEPTS AND THE PERCEPTION OF EFFECTIVENESS OF THE COURSE IN A SECOND SEMESTER COMPUTER SCIENCE COURSE

APPROVED BY
DISSERTATION COMMITTEE:

Nell B. Dale

Lowell J. Bethel

John Werth

James P. Barufaldi

Walter Martin

ACKNOWLEDGEMENTS

I would like to thank several people for their assistance in the completion of this study. Dr. Lowell Bethel and Dr. Nell Dale, my co-chairs, for their hours of time and words of encouragement. I will always feel a deep appreciation for everything they have done for me.

I am very grateful to the other members of my committee for their time and helpful suggestions, Dr. John Werth, Dr. Jim Barufaldi, and Dr. Heather Carter.

I would like to also acknowledge Dr. Bruce Elenbogen and Dr. Bruce Maxim who allowed me to use their programs in the development of the closed laboratory activities, without them this study would have taken years longer.

I am especially grateful to Doug for his patience, love, and support throughout this long and stressful endeavor. I would like to acknowledge my mom, Jan Colby, my brother and sister-in-law, David and DeAnna Burton, and all of my friends for their words of encouragement and for being on the other end of the phone when I needed to vent my frustration.

THE EFFECT OF CLOSED LABORATORY ACTIVITIES ON THE COMPREHENSION OF FIVE CONCEPTS AND THE PERCEPTION OF EFFECTIVENESS OF THE COURSE IN A SECOND SEMESTER COMPUTER SCIENCE COURSE

Publication No. ______________

Debra Lynn Burton, Ph.D.
The University of Texas at Austin, 1992

Supervisors: Lowell J. Bethel
Nell B. Dale

The investigation studied the effects of using closed laboratories on students at a major southwest research university enrolled in the second computer science course required of computer science majors. Students are required to attend a discussion section in addition to the lecture. There were a total of ten discussion sections, half were randomly assigned to remain discussion sections throughout the semester and half to act as closed laboratories for five weeks while the selected concepts were being covered (one-third of the total weeks).

In this study four hypotheses were tested, two major and two minor, using data collected by the Burton Perception Instrument and the Burton Comprehension Instrument. The tested hypotheses were: (1) no significant difference between the two groups on comprehension of the five concepts, (2) no significant difference between the two groups on perception of the effectiveness of the course, (3) no significant difference between the students identified as less likely to succeed in the two groups on comprehension of the five concepts, and (4) no significant difference between the students identified as less likely to succeed in the two groups on the perception of the effectiveness of the course.

The results of the analyses showed that the inclusion of the closed laboratories made no significant difference on concept comprehension or perception of the effectiveness of the course. Therefore the use of closed laboratories appears to be no more effective than traditional discussion sections. In terms of the students identified as less likely to succeed in the course, the results were the same, there was no significant difference between those students who attended discussions all semester and those that attended significant difference between those students who attended discussions all semester and those that attended closed laboratories for five weeks (approximately one-third of the semester) and regular discussion sections the remainder of the semester.

Table of Contents

Abstract v
Table of Contents viii

List of Tables x

Chapter 1. 1

Statement of the Problem 2
Significance of the Study 6
Theoretical Basis 8
Definition of Terms 11
Summary 12

Chapter 2. 14

Concrete Learners and Abstract Concepts 14
Effect of Laboratories on Science and Mathematics 16
Effect of Closed Laboratories on Computer Science 23
Predicting Success in Computer Science 27
The Closed Laboratory Activities 31
Development of the Null Hypotheses 34
Summary 35

Chapter 3. 38

Sample 38
Dependent Variables 41
Independent Variables 42
Methodology 43
Instrumentation 48
    Burton Comprehension Instrument 48
    Burton Profile Questionnaire 50
    Burton Perception Instrument 52
    Closed Laboratory Acitivities 54
Pilot Study 56
Data Analysis 59
Limitations of this Investigation 60
Summary 61

Chapter 4. 62

Results of Data Analysis 62
Summary of Findings 67

Chapter 5 69

Summary 69
Results 71
Conclusions 76
Suggestions for Further Research 78

Appendix A: Consent Form 80
Appendix B: Burton Comprehension Instrument (Pretest) 82
Appendix C: Reliability Measures for Pretest 88
Appendix D: Burton Profile Questionnaire 90
Appendix E: Table of ACT/SAT Scores from The University of Texas at Austin 93
Appendix F: Closed Laboratory and Discusion Section Activities 96
Appendix G: Observation Instrument 123
Appendix H: Summary of Observations 125
Appendix I: Burton Perception Instrument 127
Appendix J: Alpha Coefficient Values for the Items 130
Appendix K: Supplemental Course Survey Questions from The University of Texas at Austin 132
Appendix L: Final Exam with Posttest Embedded . . . 138
Appendix M: Pilot Study Pretest 154
Appendix N: Reliability Data for Pilot Study Pretest 160

Bibliography 162

Vita 171

List of Tables

Table 1 Variables Investigated Relative to the Use of the Laboratory in Science Teaching 20

Table 2 Assignment of Discussion Sections to Control and Treatment Groups 40

Table 3 Cronbach's Alpha Coefficient Values 54

Table 4 Means and Standard Deviations of the Pretest and Posttest for the Control and Treatment Groups 64

Table 5 Measure of the Perceptions for the Control and Treatment Groups 65

Table 6 Means and Standard Deviations of the Measures of Comprehension for Students Less Likely to Succeed 66

Table 7 Measure of the Perception for the Control and Treatment Groups for Students Less Likely to Succeed 67

Table 8 Likelihood to Succeed Versus Passed Course 74

Chapter 1

Closed laboratories have been used successfully in association with science courses for many years. In science courses such as chemistry, biology, and physics a closed laboratory is the place where students go to experiment with the concepts presented in classroom lectures. The closed laboratory activities are structured to allow students to experience the desired outcome in a specified amount of time. Since the late 1800's science classes have included laboratories, the purpose of these laboratories has always been to allow hands-on experiences for the students. Concepts covered in class can be made more concrete by experimentation in the laboratory. The use of laboratories in the high school or college curricula is so prevalent it would be rare for anyone to question the importance of their use. Laboratory work is simply part of the science game (Kirschner & Meester, 1988). For years computer programming classes were structured as strictly lectures with out-of-class programming as the primary means of applying concepts. The computer laboratories students use to complete out-of-class programming assignments are referred to as "open" laboratories. An open laboratory activity can be thought of as analogous to a chemistry assignment in which the students are told that the beakers and Bunsen burners are available at various locations on campus and the chemicals can be obtained at vending machines (Petersen & Miller, 1985). A closed laboratory activity, as the term will be used in this study, refers to a specified assignment that must be completed within a specified time period, usually within the laboratory period of one hour, in a room of computers reserved specifically for this purpose.

Statement of the Problem

Many problems for programming students relate to program comprehension, the ability to conceptualize the abstractions of a given algorithm with the flow of program code (Ramlet & Folk, 1985). Closed laboratory activities have been proposed as one method of solving this problem. The question is, are closed laboratory activities effective in improving the comprehension of computer science concepts? In order to fully study the effectiveness of closed laboratory activities two measures will be considered, the effect of closed laboratories on students' comprehension of specific concepts covered in the course and closed laboratory activities' effect on students' perception of the effectiveness of the computer science course with which the closed laboratory activities are associated. The subjects to be used will be students enrolled in the second programming class required of Computer Science majors at a large southwest research university (Koffman, Stemple, & Wardle, 1985). The course being used in this study already requires students to attend a discussion section in addition to lecture. *This investigation will determine if there is a difference in concept comprehension and perception of the effectiveness of the course in students attending both closed laboratories and discussion sections and students attending traditional discussion sections alone.* For this investigation one group of students will participate in closed laboratory activities for five selected concepts covered in the semester of the study while a second group will attend traditional discussion sections covering the same five concepts so that both are exposed to the same material for an equal amount of time. Other than the five weeks of the selected concepts, both groups will be attending traditional discussion sections.

In order to determine any differences in comprehension between the two groups, an instrument will be developed to measure the comprehension of the five selected concepts. The five concept areas were chosen because of their importance in the computer science course: complexity, stacks, queues, recursion, and sorting. They are all concepts recommended to be covered in this course by the 1991 Computing Curriculum designed by a joint task force of the Association for Computing Machinery (ACM) and the Institute for Electrical and Electronic Engineers (IEEE). These five concepts are ones that are presented in an abstract way in lecture and therefore lend themselves to being better understood and made more concrete by hands-on experiences.

In addition to the comprehension instrument, the investigator will develop an instrument, to be given at the end of the semester, to measure the students' perception of the effectiveness of the computer science course. This instrument will be designed to measure the students' perception of effectiveness in both the lecture section and the separate discussion section/closed laboratory.

The study is also designed to measure the effect of closed laboratories on concept comprehension and perception of course effectiveness of those students identified as less likely to succeed. Research attempting to measure the effectiveness of laboratories in other science disciplines has suggested that laboratory activities can be more beneficial to students identified as less likely to succeed (c.f., Ewbank, 1971; Odubunmi & Balogun, 1991; Vance & Kieren, 1971).

The major hypotheses expressed in the null form to be tested in this investigation are as follows:

H_{01} - There will be no difference in concept comprehension in the students attending lecture with both closed laboratories and discussion sections and the students attending lecture with discussion sections alone.

H_{02} - There will be no difference in the perception of the effectiveness of the course in the students attending lecture with both closed laboratories and discussion sections and the students attending lecture with discussion sections alone.

The two subhypotheses stated in the null form to be tested in this investigation are as follows:

H_{1A} - There will be no difference in concept comprehension in the students less likely to succeed attending lecture with both closed laboratories and discussion sections and those attending lecture with discussion sections alone.

H_{1B} - There will be no difference in the perception of the effectiveness of the course in the students less likely to succeed attending lecture with both closed laboratories and discussion sections and those attending lecture with discussion sections alone.

Significance of the Study

In the paper entitled "Computing as a Discipline," (1989) it was suggested that "closed labs" should be a part of computer science classes. It was felt that three hours of laboratory a week along with the normal lecture hours would enhance the effectiveness of the classes. It was proposed that by giving assignments designed to enhance understanding of the underlying concepts of computer science, as well as demonstrating certain aspects of the hardware and software being used, students would gain a richer understanding of the subject matter. In the follow-up paper "Computing Curricula 1990: Report of the Joint Curriculum Task Force," an outline was presented to be followed, based on the "Computing as a Discipline" (1989) paper, for undergraduate programs in computer science that also recommends closed laboratory assignments. In March of 1991 the final version of the curriculum developed by a joint committee from the Association of Computing Machinery and the Institute of Electrical and Electronic Engineers (ACM/IEEE) was released. Closed laboratories were an integral part of the new curriculum especially in the lower division courses (e.g., freshman and sophomore). These very important papers show that closed laboratories were being proposed as an important component of all formal computer science education programs with apparently little or no evidence that they were effective.

Because there is no empirical evidence currently available that these potentially expensive laboratories are effective in enhancing comprehension, the investigator felt that the idea of closed laboratories was one which should not be accepted without some proof of its effectiveness in computer science. Open computer science laboratory activities which are used traditionally allow the students to experiment with and/or practice the concepts presented in lecture and to make their own discoveries.

A preliminary search of the computer science literature reveals that closed laboratories are not only being proposed but accepted as an effective enhancement to lectures (Chavey, 1991; Naps, 1990; Penny & Ashton, 1990; Roberge' & Suriano, 1991; Wenner, 1990). There are many papers on how to design and use closed laboratories as well as examples of laboratory exercises in computer science (Chavey, 1991; Penny & Ashton, 1990; Roberge' & Suriano, 1991). There are several different approaches to teaching closed laboratories. One way is just to supply a supervised laboratory setting for students to work on their outside programming assignments. Other methods include giving supplemental exercises to support the concepts covered in lecture or giving experimental exercises where students discover the outcomes for themselves. This study proposes to use a combination of the latter two methods because they are the methods followed by suggested closed laboratory activities in the 1991 ACM/IEEE Curriculum. The closed laboratory activities used in this investigation are supplementary to concepts covered in the lecture and are designed in such a way as to allow the students to discover the answers to these problems on their own with the help of a computer program supplied for this purpose.

Theoretical Basis

"Many problems for beginning programming students relate to program comprehension, that is, the ability to conceptualize the abstraction of a given algorithm with the flow of the program code" (Ramlet & Folk, 1985, p. 36). According to Mincy, Tharp, and Tai (1983, p. 106) traditional computer science classroom lectures are not a particularly effective mechanism for conveying an algorithm, and the more complicated the algorithm the less effective they are. Reasons stated are that students usually understand material at different rates and have trouble visualizing what is being discussed. Closed laboratories have been used in many disciplines such as Physics, Chemistry, and Biology to assist students in their understanding of concepts presented. This investigator proposes that the use of closed laboratories in computer science will allow students to make the abstract concepts presented in lecture more concrete by having exercises on the computer that allow the students to experiment and more thoroughly comprehend the concepts at their own pace. Traditionally the students in computer science courses look at the computer only as something to be programmed and not as an aid to learning concepts (Kurtz & Johnson, 1985).

"Material that is understood is much more likely to be remembered than material that is not understood" (Schallert, 1982, p. 15). One of the modern theories of learning is that of schema theory. Schema theory describes how knowledge is stored and updated. The theory states that all of the knowledge that people acquire is stored in the form of schema and subschema in a hierarchical relationship (Schallert, 1982). Ausubel feels that the central kernel of learning, retention, and pedagogy is the principle: "new meanings are acquired by the interaction of new knowledge with previously learned concepts or propositions" (Ausubel, Novak, & Hanesian, 1978, p.127). Ausubel stresses that the meaningfulness of new information is more dependent on how well it meshes with a particular learner's cognitive structures [schema]. Thus, meaningful learning proceeds by the process of changing one's current organizations of knowledge [schema], either by elaborating a subordinate concept or by changing the superordinate structure into which subordinate facts fit (Ausubel, 1961). In other words, in order to increase our knowledge we either change existing schema to accommodate the new knowledge or add new schemas to assimilate new knowledge. In order to change one's existing schema there has to be an incentive to facilitate the change. Through the use of closed laboratories students can explore more deeply the abstract concepts that are presented in lecture. Students are given concrete exposure and a chance to experiment (interact) which may facilitate the change in their existing schema that could not be accomplished by passively sitting through the lecture alone.

Definition of Terms

Several terms with specialized definitions are used throughout the following chapters. These terms include:

Algorithm refers to a set of rules for solving a problem in a finite number of steps.

Closed laboratory activity is an activity given to students while they are in attendance in a closed laboratory. The activity is to be completed while they are in attendance under the supervision of a laboratory assistant.

Closed laboratory refers to a room of computers that is open for set amounts of time to allow students to complete a closed laboratory activity in the amount of time they are allowed in the closed laboratory. Typically the amount of time is limited from one to two hours.

Concept comprehension in this study refers to the amount of understanding of the covered concepts.

Open laboratory refers to a room of computers that is open throughout the day to allow students to work on programming assignments at their own convenience.

Open laboratory activity refers to an assignment given to students to be completed on their own time outside of the lecture section or discussion section.

Perception refers to the students' opinion of the effectiveness of the lecture section of the course as well as the discussion sections/closed laboratories. Perception as defined in "Webster's Dictionary" is an immediate or intuitive recognition or appreciation, as of moral, psychological, or aesthetic qualities (1991).

Traditional discussion sections are sessions in addition to the traditional lecture where students' meet with a Teaching Assistant in order to discuss outside assignments, review and return exams, and work exercises with pencil and paper designed to further clarify concepts covered in lecture. On the days of the closed laboratory activities the traditional discussion sections perform a parallel activity with pencil and papaer on the same concept.

Traditional lectures are classroom presentations where the instructor presents information to the students with the purpose of identifying and clarifying the knowledge to be gained in the particular course.

Summary

This study will look at the effectiveness of adding closed laboratory activities to the traditional second course taught to computer science students. The remaining chapters can be summarized as follows: Chapter II will include a summary of the supporting literature; Chapter III will describe the research design and methodology to be used in the investigation, including a brief description of a pilot study; and Chapter IV will contain a presentation and summary of the results of the investigation. In the final chapter, Chapter V, the results of the investigation will be discussed together with the implications for computer science education, and suggestions for further research.

Chapter 2

This chapter analyzes the literature pertaining to the use of closed laboratories in traditional science and mathematics courses as well as in association with computer science courses. The effect of closed laboratory activities on students' comprehension, perceptions, and on students identified as having lower ability will be examined specifically. The rationale for decisions made in the design of this investigation, the development of the closed laboratory activities, and the development of the instruments will be established. A factor in this rationale is that most of the literature on the role of the laboratory is opinion-based rather than research-based (Blosser, 1983).

Concrete Learners and Abstract Concepts

Science by nature is a highly complex and abstract subject matter area. Students are generally not able to grasp these complex and abstract concepts without the concrete materials and possibilities for manipulation which a laboratory offers (Kirschner & Meester, 1988). Toothacker (1983) states that reports show that only one third of beginning university students are actually in the formal reasoning stage. Another third of the students can be classified as completely in the concrete, operational stage. The remaining third is in a period of transition between the two stages. If this is the case, approximately two-thirds of students making use the opportunities offered them in science courses are not intellectually ready for the concepts. At best this is a waste of time and money, at worst this is a demoralizing experience for the students. Sein and Bostrom (1989) state that concrete learners benefit from concrete presentations and abstract learners learn better from a more abstract presentation but are not hampered by a concrete presentation.

"It is a well-known fact among those in the computing profession that programming is not a spectator sport" (Weiner, 1989, p. 60). However it is hard to convince students of this fact. With lecture alone it is easy for the students to become a passive participant in the lecture, especially when abstract concepts are being presented, with laboratory activities the student becomes an active learner. Two stated purposes by McCracken (1989) of laboratory assignments are to:

-- Reinforce what has been presented in lecture, making it concrete and giving the students a chance to be sure they understand it.

-- Provide some "learning through the fingertips."
-- Raise questions that prepare the students for later lectures.

Computer science students must often be able to mentally execute "static" source code to envision "dynamic" changes in data structures. This is a difficult skill to acquire, however, the computer itself can help students develop such talents through well designed simulations used in closed laboratory activities (Kurtz & Johnson, 1985; Maxim & Elenbogen, 1987). Students often have difficulty relating an algorithm, expressed in terms of executable code, to the data structures, often shown pictorially in the text. Students must learn to mentally run the algorithm and try to visualize its effects on the data structures. Without computer simulations instructors have often had to rely on the use of chalk board diagrams. These diagrams are largely static in nature and are quite time consuming to produce during the class period (Maxim & Elenbogen, 1987).

Effect of Laboratories on Science and Mathematics

In 1969 Yager, Engen, and Snider did a study in which they compared students randomly divided into three groups: one group contained students who were to perform each task in small groups and to compare their results with each other; the second group watched the teacher or one student demonstrate the task and then discussed it; and the third group only discussed what would happen, there was no demonstration at all. The students' attitude and achievement were measured to look for differences. After an ANCOVA was used on the data collected it was found that no significant difference in achievement was detected between the three groups. The only significant difference found in their study was an increase in laboratory skills ($p < .05$). One of the conclusions of this study was that laboratories should not be considered a necessity when a well trained teacher could accomplish the same effect.

Cunningham (1946) in a review of 28 research studies published from 1918 to 1946, found that six studies favored the lecture-laboratory method of instruction over the lecture-only method, 20 favored lecture-only, and two said there was no difference. In the experiment of Saunders and Dickinson (1979) they examined the effect of three different approaches: lecture-only, lecture-laboratory, and lecture-recitation upon achievement and attitude of Biology students. An analysis of covariance was used to compare adjusted posttest means for the three treatment groups. The analysis of covariance showed significant differences between the adjusted posttest means of all three dependent variables at the .05 level.

The lecture-lab subjects scored higher than the lecture-only subjects, the lecture-recitation subjects scored higher than the lecture-only subjects on the achievement test and attitude measure. The lecture-laboratory and lecture-recitation sections showed no differences. They felt the differences were due to the laboratory experience and could be attributable to the increased instructional time, the increased extent of active student involvement in learning activities as a result of smaller class size, or possibly that because the class size was smaller the students tended to be less passive. White and Rosenthal (1974) also performed a study of lecture alone versus presenting equivalent information in part by lecture and in part by demonstrations. The data collected were analyzed in a 3(lecture, demonstration, or control) x 2(pre- and posttest repeated trials) factorial design. The analyses resulted in the lecture-demonstration method significantly surpassing the lecture-only and control group at the 0.001 level. On the attitude measure the demonstration group gave a pattern of ratings more positive than the lecture group at the 0.01 level. Similar findings were reported by two other studies in which lecture-laboratory methods were compared to lecture only (Bradley, 1968; Gunsch, 1972).

Lawrenz and Munch (1984) have found that time spent in the laboratory appears to affect student's attitudes. Hofstein, Gluzman, Ben-Zvi, and Samuel (1980) did a study on the influence of laboratories on student's perception of their classroom learning environment. They used a pre/post quasiexperimental design with two groups of students: one group was a high school chemistry class, the other group was a college chemistry class of students studying to be laboratory technicians. At the same time in the middle of the year the LEI (Learning Environment Inventory) was administered. A multiple discriminant analysis was used to determine the two groups' perceptions. The results indicated that the two groups differed in their perceptions. The future laboratory technicians perceived their learning environment to be significantly more satisfying, formal, goal-oriented, organized, difficult, democratic, less apathetic, and less competitive than the high school students. They attribute this difference to the amount and type of laboratory activities. A study on mathematics laboratories by Vance and Kieren (1971) stated that a generally held feeling about laboratories is that they promoted better attitudes toward mathematics because of concrete representations and laboratory approaches to covering concepts.

There have been many experimental studies examining the effectiveness of traditional science and mathematics laboratories. The majority of these studies have shown that laboratories have no effect on achievement or attitude. This lack of effect has been attributed to poor experimental design (Blosser, 1981). An excerpt of the summary table of results of experimental studies performed from 1930 to 1980 that was produced in a report by Blosser (1981) is in Table 1.

Table 1

Variables Investigated Relative to the Use of the Laboratory in Science Teaching

Dependent Variable	Total Studies	Favored Lab	Favored Other*	NSD**	Mixed Results
Achievement	87	8	4	72	3
Attitude	32	5	1	21	5

*Includes demonstrations and no supplemental sessions.
**No Significant Difference

Vance and Kieren (1971) have studied the effectiveness of using laboratories in mathematics and have found that the laboratory approach has been particularly effective on students with low-ability. Students were randomly assigned to three groups: a laboratory group which worked in pairs with written directions directly with physical materials; the second group watched a teacher demonstrate the concrete materials; and the third group had no exposure to concrete materials. The results of the data analysis found no significance difference in achievement between the groups, except when the students of lower ability were examined separately. This subgroup scored significantly higher in the demonstration group. The attitudes of the students in the laboratory group were determined to be significantly greater on an instrument designed by the investigator. Their overall conclusion was that the teacher demonstration method was at least as effective as the manipulative laboratory with concrete materials. Ewbank (1971) also has observed that low-achievers are often "low" because they have not had the experiences they need. He states that laboratories offer an atmosphere in which to succeed and therefore give "a critical effect on their morale" (p. 562). In a study by Odubunmi and Balogun (1991) it was determined that the addition of laboratories to the regular lectures created a significant increase in performance in the students attending the laboratory sections ($p < 0.01$), especially the low and average achievers ($p < 0.01$). Odubunmi and Balogun used two groups of intact classes in which one group the teachers taught with laboratories and the other group did not.

ANCOVA was used to take out any initial inequalities because intact classes were used to ensure any differences in achievement was a result of the methods used.

Two different dissertation studies by Boghai (1978) and Grozier (1969) have determined that laboratory activities appear to be helpful to students rated as medium to low in achievement. The study by Grozier assigned students to two large groups, one with laboratories and one without. He gave a pre- and posttest on attitudes, factual material and ability to interpret data. ANOVA was used and the results indicated that the attitudes of the students with laboratories became less positive, while the attitudes of the students without laboratories became more positive. It was also found that students who scored below the median on the pretest acquired significantly more factual material than those who scored above. Overall the inclusion of laboratories was not a significant factor in factual material learned. Boghai (1978) did a slightly different study. He looked at the difference between two groups where one had discussion before laboratory and the other had laboratory before discussion. The laboratory first approach was found to be superior, especially for students identified as low aptitude. Another finding of this study was that the Teaching Assistants conducting the laboratories did not have an effect on the treatment.

In summary, the effect of laboratories on achievement and attitudes of students in science and mathematics courses is not clear. Many research findings have contradictory results. It is clear from these research findings however that establishing the effectiveness is something researchers are striving to achieve. This investigation will measure the effect of laboratories on achievement (concept comprehension) and attitude (perception of effectiveness of course) on computer science courses. Because there are many studies which have shown that students identified as lower ability or less likely to succeed are positively effected by laboratories, this investigation will include hypotheses specifically aimed at this group.

Effect of Closed Laboratories on Computer Science

There has been little research to evaluate the effect of closed laboratories to see if they will produce a significant difference in performance by the students enrolled in introductory computer science classes. After a thorough search of the literature, attending several conferences and workshops, and discussing the topic with many researchers in the field, one study was found. This study by Peterson and Miller (1985) used a different interpretation of closed laboratories than is planned for this study or as prescribed by "Computing Curricula 1991". It used the closed laboratory to perform what would have normally been outside programming activities (open laboratories). In the experiment by Peterson and Miller (1985) using closed laboratories meant there were no outside assignments; all programming was done in the closed laboratory. A major weakness of the study was that the investigators ran the experiment serially rather than in parallel. They used open laboratories (students doing programs on their own time) in the Fall semester and they used closed laboratories (students doing their programs in the laboratory at specified times) in the Spring creating dissimilar conditions for the two semesters. There were also different programming assignments given for the two semesters which is noted as a possible weakness by the author. The main purpose of the study was to find out if the organization of the laboratory had any effect on any of the independent variables, (exams, programs, and attendance). This experiment did show significant improvement in the performance of the students on programs ($p < 0.01$) between the two semesters and concluded that the closed laboratories (as they defined them) were the primary cause factor for this difference. The primary type of evaluation of the effects of closed laboratory activities in the computer science literature appears to be anecdotal evaluations of student attitudes toward the laboratory experiences. Chavey (1991) of Beloit College has developed and used closed laboratories for CS1 over the last two years and reports the following conclusions based on surveys: instructors felt the laboratories were effective, students felt the laboratories were effective, the best concept-driven laboratories were those which had introductory material presented in lecture before-hand, and students felt that their frustration with programming assignments was reduced. Weiner (1989) also reports the institution of supervised laboratories to have been a positive, rewarding experience for both faculty and students based on end of semester critique forms. He has converted all of the programming intensive courses to include compulsory supervised laboratories. Penny and Ashton (1990) state that laboratories encourage students to work together and they feel students learn more working in groups. They found experiments where students have to "find out" are particularly effective for group study. Their statements are based on answers to course questionnaires designed to assess the merits of this way of teaching. Weaver (1981) reported students judged the course content as "exceptionally good" by anonymous evaluations of a course which included closed laboratory activities. He states that it is too soon for him to judge quantitatively the effect of laboratory-based versus lecture-based courses. Where laboratory-based refers to a course heavily dependent on laboratory experiences, the lectures are written to support the laboratories. He expresses that their, his and his colleagues, "experience thus far supports their premise that the undergraduate course is certainly more enjoyable, and probably more educationally effective, than its lecture-based predecessor; an actual performance study of students taught each way has yet to be completed" (p. 136). Improved student perceptions were reported by Olson (1983) as measured by student evaluations of using closed laboratories in a beginning programming course. Olson and his colleagues have added laboratory sections to give students as much algorithm writing practice as possible. He states, "Through the lab experience the concepts of the lecture portion of the course are enhanced by providing several exercises for each of the concepts and by involving the student in the exploration of the key programming constructs" (p. 165). Wilson (1991), and Wilson and Trenary (1981) also report very positive experiences with the inclusion of required laboratory activities as gathered from student evaluations.

Predicting Success in Computer Science

Campbell and McCabe (1984) did a study in which they determined that students who persisted with computer science majors or switched to engineering or another science, differed from students who switched to other fields in terms of SAT scores, high-school rank, and high school math and science background. A discriminant analysis of their data determined that SAT scores, high school mathematics and science background, and gender were the most predictive. It is explained that they recognize gender is not an achievement indicator but it consistently appeared as a variable in the classification models. They feel this observed sex-related difference in persistence may reflect a reaction to the demands of the major. Science and engineering are exceptionally demanding majors requiring many hours of study and attendance in addition to attending lecture that many students are unwilling to devote.

Gathers (1986) studied ten factors to determine their relationship to success in the first computer science course for majors. The ten factors studied were: high school grade point average, ACT scores (English, math, social science, natural science, and composite), Nelson Denny Reading scores (vocabulary, comprehension, and total grade equivalent), and math placement test score. The significant factors in the successful placement of college freshman were the ACT English and a math placement test score. Ramberg and Van Caster (1986) did a study to predict success in computer science courses. They found that the single most important key to success in computer science is a prior exposure to computers, whether that be a literacy/programming course in high school or college. It also appeared that prior math background was a good predictor in many cases.

Wileman, Konvalina, and Stephens (1981) found that there were significant differences ($p < 0.001$) in students who withdrew from a computer science course and those who did not. The causes for the differences all related to the students' mathematical background, high school performance, previous computer science education, and the number of college mathematics courses taken. Dey and Mand (1986) also found that background skills in mathematics (college and high school) appear to be contributory to a student's perceived success, as reported on a survey, in an introductory computer science course.

Butcher and Muth (1985) found that Math ACT scores and high school grade point average can be used to predict success, making a 'C' or better, in introductory computer science classes. A study by Taylor and Mounfield (1989) found that prior computer science course work, either high school or college, was a major factor for success in an introductory computer science course. The investigators surveyed students enrolled in a computer science course and compiled these results for those that were successful and unsuccessful (success is defined as making a 'C' or better in the course). Other factors found in the Taylor and Mounfield study were:

- -- students who worked part-time up to 20 hours were more successful than students who did not work or worked more than 20 hours
- -- males were more successful that females, females were found at the extreme ends of the spectrum, they either did very well or very poor
- -- freshman were significantly more successful than students in any other classification.

Werth (1986) found a significant relationship ($p < .05$) between the letter grade in the first computer science course and the students' college grades, the number of hours worked and the number of high school mathematics classes.

In developing the Burton Profile Questionnaire the ethnicity of the students was also considered. None of the previous studies cited specifically addressed the issue of ethnicity, yet a study of pre-college students by

Martinez and Mead (1988)), states that when racial/ethnic groups were compared White students were shown to have a clear advantage over African-American and Hispanic students across the grades in computer competence. This report compared students in the third, seventh, and eleventh grade, which would indicate that this difference should be a consideration in college computer courses as well. This predictor is being used in this investigation because very few minority students are successful in computer science.

In summary the factors that were found to predict success in computer science courses which were subsequently used to create the Burton Profile Questionnaire are: prior computer science courses in high school; prior computer science courses in college; prior mathematics courses in high school; prior mathematics courses in college, ACT/SAT math, english and composite scores; hours worked per week; high school grade point average; college grade point average; classification (freshman, sophomore, junior, senior, or graduate); and ethnicity. These factors represent measures of students maturity level, prior knowledge, and commitment.

The Closed Laboratory Activities

In the report entitled "Computing as a Discipline" (Denning et al, 1989) the following recommendations are made:

-- computer science should model the traditional physical sciences and engineering which include closed laboratory activities

-- the laboratory sessions should supplement, not replace, the usual programming and other written assignments

-- closed laboratory activities should be carried out under the guidance of a lab instructor

-- closed laboratory activities should be self-contained in the sense that an average student should be able to complete the work in the time allocated

-- the closed laboratory activities should encourage students to discover and learn things for themselves.

The "Recommended Curriculum for CS2, 1984" (Koffman et al, 1985), identifies the concepts to be covered in the course being used for this experiment. The following topics are specified:

-- software engineering

-- review of data types

-- data structures, which includes stack and queues

-- implementation of data structures, which includes the implementation of stacks and queues

-- recursion

-- searching

-- analysis of algorithms using "big-O', which is referred to as "complexity" in this investigation

-- sorting.

CS2 is recommended to be taught either as a standard lecture class with one instructor and one assistant or as a large lecture class with one instructor and several assistants, each of whom has responsibility for discussion sections with a subgroup of the class.

Included in "Computing Curricula 1991" are specific recommendations for closed laboratory activities to be used in the second computer science course. When covering the concept of recursion in lecture, the suggested closed laboratory activity is to "trace the execution of a recursive program, viewing changes of values of local variables and parameters" (p. 41). When the lecture topic is complexity classes, the suggested closed laboratory is "timing of selected algorithms from various complexity classes, in order to see difference in running times for small and large problems" (p. 42). "Corroboration of theoretical complexity of selected sorting algorithms by experimental methods, identifying differences among best, average, and worst case behaviors" (p. 43) is the suggested closed laboratory activity when the lecture topic is sorting. There is no closed laboratory activity suggested when the concepts of stacks and queues are being covered in lecture; therefore this investigator developed an exercise to allow the students to experiment with these data structures. One form of closed laboratory activity that is suggested by "Computing Curricula 1991" is "the execution of hardware, software, or simulators to observe some phenomenon, either by data collection or by visualization" (p. 19). This was the form of closed laboratory activity chosen for this investigation. An article by Maxim and Elenbogen (1987) describes programs they developed for demonstration in classroom lectures which incorporate the ability to view simultaneously the data structure model, its virtual machine representation, the symbolic execution of the programming algorithm, and the dynamic changes in the data structure effected during the execution of the algorithm. This is a set of programs which include graphical demonstrations of recursion, sorting, stacks and queues. There were no programs for demonstrating complexity available, therefore this investigator developed one for this purpose. Four of the six programs used for the closed laboratory activities were supplied by Maxim and Elenbogen.

Development of the Null Hypotheses

Any research done in relation to closed laboratories in computer science has been primarily restricted to that of surveys and course evaluations. Research in the area of traditional laboratories as used in science and mathematics has not shown conclusively that laboratories are effective, in terms of achievement and attitudes, except in the performance of low-achievers. Low-achievers are believed to benefit from laboratories because of the hands-on aspect and the fact that they can work at their own pace, repeating the exercise as many times as time will permit.

Based on these research findings this carefully designed investigation will attempt to determine if closed laboratories used in computer science classes will have any effect on concept comprehension and students' perception of effectiveness of the course. With the numerous studies stating that laboratories have been shown to be effective on those students identified as low achievers, this investigation will also attempt to determine the effect of closed laboratories on concept comprehension and perception of the effectiveness of the course for students identified as less likely to succeed.

Summary

The review of the literature has been used to develop the rationale of this investigation. Closed laboratories are being proposed for use in the computer science curriculum because they allow students to make concrete the abstract concepts covered in lecture and because they allow students to learn the concepts at their own rate. Because computer science is such a young field it has little historical research in this area. The studies in the traditional sciences, biology, physics, chemistry, etc. on the effectiveness of using laboratories are not conclusive. There are many contradictory findings in the results of these studies. Many of these studies contain weaknesses in the designs which this investigation will attempt to overcome, such as the Hawthorne effect, the absence of a control group, and using instrumentation which is not shown to be reliable or valid. Because the effectiveness of laboratories has not conclusively been shown in the traditional sciences, they should not be accepted without question in computer science. The existing literature does reveal a common finding in several studies which relate to students of lower ability as identified by several different methods. The laboratory approach does appear to have a positive effect on students of lower ability. It is proposed that this effect is caused by the hands-on environment of the laboratory; because the students are working on their own and can therefore go as slow or as fast as they wish, they can repeat the exercise until they understand it, and have positive feedback by being able to figure out the correct answers on their own. Based on this previous research a planned comparison is being placed into this study to examine the effects of closed laboratories on those students identified as less likely to succeed.

Identifying students less likely to succeed in computer science is a very popular topic. Computer science traditionally has a high failure rate in the beginning classes. There are many articles which propose to have found the predictors of success. This investigator has examined these studies and come up with the predictors common to all and that apply to the sample being used in this study. All of the identified predictors can be tied directly to the level of maturity of the students, their prior knowledge, and their ability to be responsible.

The computer science literature has been reviewed to determine what the important concepts are and which should be used in this study, as well as the associated closed laboratory activities to accompany these topics. The "Computing Curricula 1991" and "Recommended Curriculum for CS2, 1984" both proposed that the same concepts be covered. Therefore this investigator and the other instructor, who have combined teaching experience of 30 years, chose topics which are important and would lend themselves to closed laboratory activities. "Computing Curricula 1991" also had suggestions for closed laboratory activities which would be appropriate for these topics. Therefore the concepts and closed laboratories chosen for use in this investigation are valid and supported by the literature.

In the next chapter (III) the research design will be described together with the techniques, instruments, and statistical analyses to be used.

Chapter 3

This chapter describes the population and the sample selected from it, the dependent and independent variables, the development and pilot testing of the instrumentation, and the closed laboratory activities developed. It also describes the overall investigation, data collection methods, and the statistical analyses that were employed. A short summary of the pilot study is also included for background information.

Sample

In this investigation the sample consisted of students registered for the second computer science course, Computer Science II, required at a major southwest research institution. This course required students to attend a discussion section in addition to a lecture. There were two lecture sections each with five discussion sections, for a total of ten discussion sections. There were five different Teaching Assistants assigned to lead the discussion sections, and it was decided that Teaching Assistants should teach only one type of group -- treatment or control. However, because of the even number of discussion sections and the odd number of Teaching

Assistants, one of the Teaching Assistants, the most experienced, was chosen to lead one section of each type (treatment and control). The Teaching Assistants were briefed weekly on the topics to be covered and then were trained separately for either the closed laboratory activities or the traditional discussion sections. There were two instructors for the course, one for each lecture section (each with approximately 140 students).

In the first week of the semester of the investigation, the students were asked to sign a consent form, fill out the Burton Profile Questionnaire, and to take the pretest. Of the 288 students originally registered for the course, only 125 completed the pretest, filled out the questionnaire, signed the consent form, took the posttest, and were not repeating the course. Therefore N = 125 is the largest sample in this investigation.

The assignment of discussion sections to treatment or control group was accomplished as follows. The sections were divided into morning (before 12:00 p.m.) and afternoon (after 12:00 p.m.). In the morning sections it was decided every other one would be either control or treatment, in order to balance time-of-day. A coin was flipped to decide which would be control (tails - Wednesday 8:00, Wednesday 11:00, and Thursday 10:00; heads

- Wednesday 9:00, Thursday 8:30, and Thursday 11:00). The coin came up tails. In the afternoon it was decided that the Wednesday sections should be one control and one treatment, with the opposite assignment on Thursday. Therefore heads was assigned to Wednesday 12:00 - 1:00 being control and Thursday 2:00 - 3:00 being treatment, tails was the opposite control and treatment assignment. A coin was flipped and came up heads. The final results are summarized in Table 2.

Table 2

Assignment of Discussion Sections
to Control and Treatment Groups

Discussion Section	Group	TA (0 - 4)
Wednesday (instructor 0)		
8:00 - 9:00	Control	0
9:00 - 10:00	Treatment	1
11:00 - 12:00	Control	2
12:00 - 1:00	Control	2
1:00 - 2:00	Treatment	3
Thursday (instructor 1)		
8:30 - 9:30	Treatment	1
10:00 - 11:00	Control	0
11:00 - 12:00	Treatment	4
2:00 - 3:00	Treatment	4
5:00 - 6:00	Control	3

One instructor ended up with three control and two treatment sections, while the other instructor ended up with two control and three treatment. This balanced distribution had the result of taking instructor bias out of the end results; therefore instructor was not considered as a factor in the statistical analyses. An investigator prepared questionnaire was given to all of the students to determine whether they should be considered more likely to succeed in the course based on attributes which have been found to be representative in the literature (see Chapter II). These attributes include mathematics background, scores on the SAT or ACT, previous experiences with computer science, and the number of hours worked per week (c.f., Taylor & Mounfield, 1989; Werth, 1990). Based on the results of this questionnaire students were divided into two groups, those more likely to succeed and those less likely to succeed.

The sample (N = 125) consisted of computer science students who were entering their second college level computer science course. These students were representative of a population of second semester students taking computer science at a very large research university.

Dependent Variables

One dependent variable in this experiment was the students' comprehension of five selected concepts in computer science. This comprehension was measured by an instrument developed specifically for this purpose, to be referred to henceforth as the Burton Comprehension Instrument.

The second dependent variable that was measured for this experiment was the students' perception of the effectiveness of the course. The students were asked to assess the lecture section as well as their discussion section/closed laboratory. This was measured by an instrument developed specifically for this study, to be referred to as the Burton Perception Instrument. The development and pilot testing of these two instruments are described in the section entitled "Instrumentation".

Independent Variables

The independent variable for this investigation is the instructional approach used to teach the supplemental discussion sections associated with the course. The difference in the two groups of students was that during the five weeks of the semester when the selected concepts were being studied, the treatment groups participated in closed laboratory activities while the control groups were instructed on the specified concepts through the traditional discussion section method. The closed laboratory activities involved allowing the students to determine the answers to provided exercises through their own interactions with supplied computer programs. The duties of the Teaching Assistant in the closed laboratory were to explain the activity to be performed and how to run the associated program, and to answer questions about the activity or problems with the hardware or software in the laboratory room. The control groups were given the opportunity to work out the answers on their own with pencil and paper and then after 15 or 20 minutes of elapsed time the correct solution was discussed by the Teaching Assistant.

Methodology

During the first week of the spring semester 1992 all of the students attending the second computer science course at a major southwest research institution used for this study were given a pretest, asked to sign a consent form and to fill out the Burton Profile Questionnaire. The lecture sections were taught in the traditional manner; there were no changes because of the on-going experiment. The two instructors for the course used the same syllabus, met at least twice weekly to confer on what was being taught in both classes, gave the same type of in-class quizzes, and gave the same examinations on the same nights and times. The Teaching Assistants who were assigned to the treatment groups were required to meet with the investigator to run through the closed laboratory activities in the closed laboratory room to be familiar with the activity to be performed and to be aware of any problems that could arise. The Teaching Assistants met with this investigator to practice the closed laboratories before each of the five actual closed laboratory sessions throughout the semester. The Teaching Assistants who were performing the traditional discussion sections also went through the activities with this instructor to insure the correct solution and to familiarize them with possible questions that could arise. The students had access to the office hours of both instructors and all of the Teaching Assistants. The students were informed when they signed the consent form that they were participating in an experiment but were not given any details of the experiment. All students were required to attend discussion sessions in addition to their lecture sections every week. The only difference between the students in the experimental versus control groups was that the experimental groups went to the computer laboratory for closed laboratory activities in the six weeks when the selected concepts were being covered, with the rest of their discussion sections being taught in the traditional way using paper and pencil rather than a computer. The control group had all of their discussion sections taught in the traditional method. The traditional method is for the Teaching Assistants to give the students an exercise to complete in the first 15 to 20 minutes with the rest of the discussion section being spent discussing the correct method for working the exercise and the concept being covered.

The students were required to attend every discussion section throughout the semester. To insure students' attendance a policy was imposed which deducted five points from the student's overall course total for each absence. Students were encouraged to attend their Teaching Assistant's other section if they had to miss their assigned time. If they attended another section, they would not have points deducted.

The posttest was administered as part of the final exam. The final had a total of 77 objective questions of which 25 were the investigation posttest questions. The posttest questions were interspersed within the other questions on the final in order to preserve continuity of the concepts in the final exam questions. A copy of the final examination with the posttest questions marked by an asterisk (*) is included in the appendices.

The effectiveness of the closed laboratories is tied directly to the performance of the Teaching Assistants and what they did in the discussion sections. The Teaching

Assistants had been trained on how to conduct their respective duties in closed laboratory and discussion sections, but as was determined in the pilot study, this does not insure that they were performing as they were instructed. In order to validate that the Teaching Assistants were performing as they were instructed an observer was asked to monitor their performance. The observer was asked to attend one section of each Teaching Assistant's discussion sections or closed laboratory sections twice during the semester of the investigation. The weeks selected for the observations were ones in which the closed laboratory activities were taking place in parallel with the traditional discussion sections. The observer, even though having been briefed on the different duties of the Teaching Assistants in discussion section versus closed laboratory, was not informed which were the control groups and which were the treatment groups. The Teaching Assistants were not told that they were going to be observed. The observer chosen has 15 years experience as a secondary classroom teacher and three years experience as a Teaching Assistant at the major southwest research university used in this investigation. In addition, she has gone through special training in a Supplemental Instruction program designed to make Teaching Assistants better at helping students not only learn and understand difficult material but also to learn to study and take exams in their subject areas. When selected to act as an observer she had just completed months of research on the topic of conducting observations and had recently completed an investigation of her own. This investigator met with the observer frequently to develop an appropriate observation instrument and to brief her on what was expected of the Teaching Assistants in traditional discussion sections and in closed laboratories. Upon discussing the best approach to design an observation instrument for this particular purpose, this investigator designed a draft version for the observer to use. The observer attended a discussion section with this instrument and determined it was too detailed and another, less detailed but more effective instrument, was finally designed. Based on her experience using the first one, she felt satisfied with the final version. The observation instrument used for this study contained five questions in a Likert-scale format relating to the Teaching Assistants' ability to explain the activity to be performed, ability to lead discussions by asking questions of students, ability to answer students' questions satisfactorily, and their enthusiasm and responsiveness toward the students. A copy of the observation instrument used is included in the appendices.

The results of the observations will be used to explain any differences in the students' measures of comprehension or perception between Teaching Assistants if they in fact do occur.

Instrumentation

Burton Comprehension Instrument

The Burton Comprehension Instrument was developed over a period of two semesters. During the summer of 1991 the investigator determined that in order to measure the comprehension of the students for the course used in this investigation an accepted instrument had to be developed. After searching the literature it was determined that there really was no specific instrument for measuring these exact concepts: complexity, stacks, queues, recursion, and sorting. However, the Educational Testing Service administers Advanced Placement exams for high school students seeking college equivalency credit for the course used in this study. The investigator was able to obtain three versions of these exams, 1984, 1988 and 1990. Questions from these exams which covered the identified concept areas: recursion, complexity, sorting, stacks, and queues, were placed on the final exam for the summer of 1991. Fifteen of these questions which were determined to have the highest reliability rating were selected for inclusion in the pretest used in the pilot study during the Fall of 1991. After carefully evaluating the pretest used for the pilot study this investigator determined that it was not adequate to measure the students comprehension of the concepts. Therefore more questions were developed by this investigator and the other instructor, and they were pilot tested on the final exam during the Fall of 1991. Twenty-five questions were selected that became the pre- and posttest for the present study. The distribution of the number of questions per concept is:

-- five questions on complexity
-- nine questions on stacks and queues
-- five questions on recursion
-- six questions on sorting

The overall Alpha reliability of the questions is 0.76, the item by item coefficient of correlation measures are included in the appendices. The reliability of a test refers to its internal consistency, that whatever it is measuring, it does so consistently upon each administration. The item coefficient of correlation indicates the item's ability to discriminate between those students who score high and those who score low on the test as a whole.

Burton Profile Questionnaire

Based on an extensive review of the literature related to identifying success in beginning computer science courses, a list was made of the characteristics that correlate with success. Scores from the SAT/ACT, the number of previous computer science courses in college, the number of previous computer and mathematics classes in high school, college grade point average, the number of hours worked per week, and ethnicity were used to identify students less likely to succeed (see Chapter II). A questionnaire was developed related to these characteristics. The Burton Profile Questionnaire was administered to the students in the first week of the study in order to identify students as either less likely or more likely to succeed based on their voluntary responses to the questions. The questionnaire contains 19 questions where the levels used to distinguish the less likely from the more likely were taken from the literature except in the case of the SAT and ACT scores. The SAT and ACT scores were only eluded to in the articles referenced therefore this investigator obtained the levels used by the university where this study was performed to determine the levels at which it could be concluded a student would be more or less likely to succeed. A copy of this table is included in the appendices along with a copy of this questionnaire.

The Burton Profile Questionnaire was administered during the first week of the study and was subsequently used to determine those students who could be identified as less likely to succeed. After examining the responses to the questionnaire it was determined that two of the factors identified by the literature could not be used on this sample. Of the N = 125 students in the investigation 120 were either not working or working more than 20 hours a week and less than 60 students identified themselves as freshman, both of which were identified as predictors of success at the institution used in the study cited in the literature (Taylor & Mounfield, 1989). Therefore these two factors were removed from consideration when assigning students to the less likely to succeed group. The questionnaire asked the students to identify their gender. After carefully reviewing the literature and noticing a trend in the sample it was decided that gender alone was not a valid predictor. Females tend to be in the very high achievers or the very low achievers, therefore the other predictive factors could identify into which of these groups they should be classified (Taylor &

Mounfield, 1989). After these adjustments to the questionnaire it was determined that there were 57 students identified as less likely to succeed.

Burton Perception Instrument

In researching the literature to find an instrument to measure the perception of the students for this investigation, the investigator determined there was no existing instrument that could be used. The perception of the student as defined for this study refers to how the student rates the effectiveness of the lecture and discussion/closed laboratory section portions of the course. The Measurement and Evaluation Center at The University of Texas at Austin supplies supplemental questions for instructors to use in conjunction with their regular course evaluation instruments (included in the appendices). These pre-selected questions were reworded slightly to adapt them to the course used for this investigation and resulted in the Burton Perception Instrument. All of the 25 questions on the instrument are worded in the positive and are in a Likert-scale format. The students were asked to assign a value to a phrase ranging from strongly agree to strongly disagree. After developing the questionnaire, copies of it were given to seven professors in the Computer Sciences Department at the large southwest research university forfeedback. None of them suggested any changes and felt that the instrument would allow students to properly assess the course.

Schriesheim, Eisenbach, and Hill (1991) did a study in which they administered three similar scales to 522 students. The scales consisted of either regular, negated, or a combination of both types of items. The results indicated that the subjects had difficulty expressing agreement by disagreeing with the negated items. Therefore they concluded that it makes little sense to employ negated items to control for subject agreement response tendencies. In fact, they question why item reversals are still advocated. Thus it was decided not to include negative questions.

This instrument was used in the pilot study during the fall of 1991, and a statistical analysis was performed to determine its reliability. The first 16 questions dealt with the lecture section, the last nine referred to the discussion section. The statistical results are in Table 3.

Table 3

Cronbach's Alpha Coefficient Values

Section	Questions	Alpha Coefficient
Lecture	1 - 16	0.759
Discussion	17 - 25	0.824
Total	1 - 25	0.833

The alpha coefficient values for the individual questions as well as a copy of the Burton Perception Instrument are included in the appendices.

Closed Laboratory Activities

The five concept areas to be included in the closed laboratories were complexity, recursion, stacks, queues, and sorting. These five concepts were selected because they represent five of the major areas that students are expected to have learned when they finish the course (see Chapter II). The laboratory activities allowed the students to experiment with the appropriate concepts by running programs that were provided for that purpose. The laboratory on *complexity* required the students to run a program that allowed them to determine the complexity of two algorithms performing the same task where the two algorithms are of different complexity classes. There are four pairs of such algorithms with the goal that by the time the fourth pair is executed the students will have developed some skill at determining relative complexity and will be able to answer questions concerning these values with increasing effectiveness. The laboratory activities on *stacks and queues* and *recursion* provided the students with programs that graphically demonstrated the execution of these concepts. The goal was to make the abstract ideas of stacks, queues, and recursion more concrete by allowing the students to "see" the consequences of actions. The program used included computer graphics so that they could literally see the changes in the structures. They were given an exercise to complete while experimenting with these concepts that were designed to help guide their learning. The closed laboratory activity on *sorting* had the students run a program that allowed them to execute different sorting algorithms. They were asked to determine which sorts are best for specified types (already sorted, nearly sorted, random, and sorted in reverse) and sizes of input, both of which cause different sorting algorithms to be superior. Copies of the closed laboratory activities and the corresponding discussion section exercises are included in the appendices.

Students come into the course with knowledge of different computer systems. This experiment was conducted on AT&T computers. In order that there be no interference caused by unfamiliarity with using these computers, an introductory laboratory assignment was developed which was on *program testing*. The major objectives of this assignment were to: (1) allow the students to learn where the computer laboratory was located, (2) allow the students to become familiar with the computers to be used, and (3) allow the Teaching Assistants to practice administering closed laboratory activities with students in the room. Up to this time the Teaching Assistants assigned to do the closed laboratory activities had only worked the activities themselves on a computer with only this investigator and the other Teaching Assistants.

Pilot Study

The pilot study was conducted during the fall semester of 1991. There were two lecture sections, one lead by this investigator and the other by a second instructor, each with four discussion sections for a total of eight sections (N = 125). The decision was made to have each Teaching Assistant lead one treatment and one control group. A pretest was given which contained 15 questions that had been pilot tested the previous summer session and were found to have high reliability ratings (included in the appendices). The Teaching Assistants met with this investigator and the other instructor weekly to discuss the topics to be covered each week in the discussion sections. They were briefed on the differences between how to conduct the discussion section and the closed laboratory groups. No observations of their performance in the discussion sections were made because the investigator assumed that the Teaching Assistants would do as they had been instructed. After informal discussions with the students, this was determined not to be the case. The final exam for that semester contained the 15 item questionnaire from the pretest as well as a number of other questions which were being pilot tested for inclusion in the actual study that was to take place the following semester. No statistical analyses were performed on the data gathered from this study because of a major limitation. After the decision had been made to conduct this study it was discovered that the Computer Sciences department at the research institution being used during the fall of 1991 did not have a computer room with enough computers to perform closed laboratory activities. This shortage of computers meant that students had to be split up within the treatment discussion sections, some of them used the small computer room in the Computer Sciences Department, with 15 computers, and the rest used a computer room in the College of Education. The splitting between different computer rooms caused the students to be splintered into unequal groups, some as large as 30 while others were as small as six, making it impossible to run any meaningful statistical analyses. Another complication caused by the shortage of computers and each treatment discussion section being split between the two computer rooms, which were in different buildings on opposite sides of campus, was that a portion of the students had their closed laboratory activities with a Teaching Assistant different from the one they had the remainder of the time.

This pilot study resulted in the strengthening of the research design used in the present investigation. It was determined that 15 questions were not enough to adequately measure the students' comprehension of the concepts under study. The laboratory activities were also enhanced because of this pilot study. It was determined that two of the closed laboratory activities were too long, and another was given at the wrong time in association with the lecture causing confusion and the laboratory activity to be ineffective. The pilot study allowed the investigator to strengthen the present investigation in terms of the research design and the closed laboratory activities, as well as revealing the importance of the association between the timing of the presentation of lecture topics and completing the related closed laboratory activity.

Data Analysis

This investigation was a pretest-posttest control group design. After a careful review of data analysis methods used in research related to this study (see Chapter II) the investigator decided to use ANCOVA to determine differences in concept comprehension. The pretest was used as a covariate to take out any differences in comprehension of the selected topics that may have been present at the beginning of the semester. The investigator realizes that the "homogeneity of regression" assumption must be satisfied for the ANCOVA to be valid. This assumption requires that the correlation between the covariate (pretest) and the dependent variable (posttest), when computed separately within each group, is essentially the same across all groups (LaGaccia, 1991). Based on previous research (Ewbank, 1971; Vance & Kieren, 1971), a significant improvement in performance in students identified less likely to succeed could be expected, and therefore a planned comparison was placed into the design. For any particular comparison, a test is more powerful when planned than when done post hoc (Hays, 1988).

An ANOVA is planned for the data gathered with the Burton Perception Instrument used to measure the students' perception of the effectiveness of the lecture and closed laboratory/discussion sections. The responses of the students less likely to succeed in the control groups and the treatment groups will also be compared.

The number of students less likely to succeed in each group will be examined for any unusual distribution trends before forming any interpretations of the results.

Limitations of this Investigation

One of the Teaching Assistants failed to attend the first discussion section which was the section in which the students were pretested, signed the consent form, and filled out the Burton Profile Questionnaire. The students were told to come in on their own time to perform these tasks. Enough students came in on their own to allow this section to be included in the study, but it was the smallest in number of students. Students dropping the course, failing to fill out a consent form, being dropped from the study because they were repeating the course, and/or failing to take the pretest caused the total number of students in the experiment to drop from N = 288 to approximately N = 125. Statistical significance is particularly sensitive to sample size. When a sample size is small, the possibility of a Type II error (failure to reject a false null hypotheses) is greater than when the sample size is large (LaGaccia, 1991).

Another limitation of this investigation is that the closed laboratory activities were only performed in five weeks of the semester. This amounts to only one-third of the weeks of the semester being devoted to closed laboratory activities. Therefore the effect of closed laboratories on students' perception of the effectiveness of the course will be difficult to detect.

Summary

In this chapter a review of the research design has been presented, including the development of the instruments and closed laboratory activities, a brief summary of the pilot study, and the identification of the limitations of the investigation. The results of the study will be summarized in Chapter IV.

Chapter 4

The Burton Perception Instrument was administered on the same day as the required course evaluations. The day for course evaluations to be administered was included in the course syllabus, therefore students who were not interested in filling out a course evaluation failed to attend class. This unfortunate circumstance resulted in only 92 of the students being eligible to participate in the investigation after completing the Burton Perception Instrument, thereby further reducing the sample size for testing the hypotheses dealing with the students' perceptions.

Results of Data Analysis

A SAS program on an IBM mainframe at The University of Texas at Austin was used to perform the analysis of the data collected. The instruments used to measure the dependent variables, concept comprehension and perception, were developed and pilot tested prior to the investigation. The details of the pilot testing and reliability information of the instruments are included in Chapter III. An analysis of covariance was chosen to analyze the data provided by the Burton Comprehension Instrument to test the first major null hypothesis:

Hypothesis 1: There will be no difference in concept comprehension in the students attending lecture with both closed laboratories and discussion sections and the students attending lecture with discussion sections alone.

In order to appropriately use an analysis of covariance (ANCOVA) the "homogeneity of regression" test must be performed. This test is run to show that the correlation between the covariate, pretest, and the dependent variable, posttest, when computed separately within each group, is essentially the same across all groups (LaGaccia, 1991). This test was run and no significant difference was found, therefore the null hypothesis was not rejected.

Table 4 contains the means and standard deviations on the pretest and posttest measures of comprehension for the treatment and control groups. The results of the ANCOVA ($F = 0.13$) was that no significant difference was found at the .05 level between the control and treatment groups' performance on the posttest. Therefore Hypothesis 1 was not rejected.

Table 4

Means and Standard Deviations of the Pretest and Posttest for the Control and Treatment Groups

		Pretest		Posttest	
group	N	Mean	Standard Deviation	Mean	Standard Deviation
control	64	6.94	4.08	20.35	2.81
treatment	61	7.16	3.79	20.54	2.99

The second major null hypothesis dealt with the students' perception of the effectiveness of the course. In order to test this hypothesis the students had to have filled out the Burton Perception Instrument. Due to absences on the day that the instrument was administered there were only 92 students included in this analysis (N = 92).

Hypothesis 2: There will be no difference in the perception of the effectiveness of the course in the students attending lecture with both closed laboratories and discussion sections and the students attending lecture with discussion sections alone. Table 5 contains the means and standard deviations of the perception measures for the treatment and control groups. The ANOVA for Hypothesis 2 (F = 0.84) found there was no significant difference at the .05 level between the control and treatment groups' perception of the course. Therefore Hypothesis 2 was not rejected.

Table 5

Measure of the Perceptions
for the Control and Treatment Groups

group	N	Mean	Standard Deviation
control	47	59.62	12.68
treatment	45	60.29	13.93

The third null hypothesis, Subhypothesis 1, was tested using analysis of covariance (ANCOVA). The "homogeneity of regression" analysis was performed for this data also and again the null hypothesis was not rejected. The sample size for this analysis was N = 57 due to the test being run only on those students identified as less likely to succeed.

Subhypothesis 1: There will be no difference in concept comprehension in the students less likely to succeed attending lecture with both closed laboratories and discussion sections and those attending lecture with discussion sections alone.

Table 6 contains the means and standard deviations on the pretest and posttest measures on the Burton Comprehension Instrument for the treatment and control groups. The results of the ANCOVA (F = 0.08) showed that there were no significant differences at the .05 level between the control and treatment groups' performance on the posttest. Therefore Subhypothesis 1 was not rejected.

Table 6

Means and Standard Deviation
of Measures of Comprehension
for Students Less Likely to Succeed

		Pretest		Posttest	
group	N	Mean	Standard Deviation	Mean	Standard Deviation
control	34	6.53	3.96	19.91	2.96
treatment	23	5.65	2.65	19.74	3.24

The fourth null hypothesis, Subhypothesis 2, deals with the perception of the effectiveness of the course for those students identified as less likely to succeed. In order to test this subhypothesis the students had to have filled out the Burton Perception Instrument. Due to absences on the day that the instrument was administered and the fact that there were only 92 students identified as less likely to succeed there were only 43 students included in this analysis.

Subhypothesis 2: There will be no difference in the perception of the effectiveness of the course in the students identified as less likely to succeed attending lecture with both closed laboratories and discussion sections and the students attending lecture with discussion sections alone.

Table 7 contains the means and standard deviations of the perception measures for the treatment and control groups of students identified less likely to succeed. The results of the ANOVA for Subhypothesis 2 (F = 0.86) found there was no significant difference at the .05 level between the control and treatment groups' perception of the course. Therefore Subhypothesis 2 was not rejected.

Table 7

Measure of the Perception
for the Control and Treatment Groups
for Students Less Likely to Succeed

group	N	Mean	Standard Deviation
control	24	55.04	11.46
treatment	19	57.58	12.76

Summary of Findings

This investigation tested two major hypotheses and two subhypotheses relating to student concept comprehension and perception of the effectiveness of the course. Student concept comprehension as was measured by the Burton Comprehension Instrument was not significantly different between the control and treatment groups so the first major null hypothesis was not rejected. Students' perception of the effectiveness of the course as measured by the Burton Perception Instrument was not significantly different between the control and treatment groups so the second major null hypothesis was not rejected.

The two subhypotheses dealt specifically with the group of students identified as less likely to succeed. Student concept comprehension as was measured by the Burton Comprehension Instrument was not significantly different between the control and treatment groups for the students less likely to succeed so the first subhypothesis was not rejected. Students' perception of the effectiveness of the course as measured by the Burton Perception Instrument was not significantly different between the control and treatment groups for the students less likely to succeed so the second subhypothesis was not rejected.

Chapter 5

This chapter is a summary of the investigation, a discussion of the results, implications of the results and conclusions are included. Also, suggestions for future research are listed.

Summary

The present study examined the effects of using closed laboratory activities in addition to traditional classroom lectures in computer science. Students (N = 288) who were registered for the second computer science course required of majors at a major southwest research university during the spring semester of 1992 were to be used in this investigation. Due to extensive drops and incomplete information a total of N = 125 of these students were actually used in this investigation. Students who registered for this course were registered for specific discussion sections assigned to the lecture sections, for the five weeks when the treatment concepts were covered these discussion sections were randomly chosen to stay traditional discussion sections or to be used as closed laboratory sections. The closed laboratory sections participated in closed laboratory activities for five of the concepts traditionally covered in the course. A closed laboratory activity is one in which the students go to a computer room and perform an activity devised to allow them to explore an abstract concept that has been covered in lecture in order to make this concept more concrete. The closed laboratory activities are designed to be completed in the time associated with the section; for this study it was 50 minutes each. The investigator and another instructor taught the lecture sections, five Teaching Assistants led the discussion sections/closed laboratories. The Teaching Assistants were trained in administering closed laboratory activities and/or traditional discussion sections and were observed during the course of the semester. The observer was asked to verify that the Teaching Assistants were doing what they were trained to do. The results of the observations showed that the Teaching Assistants in the control and treatment groups were doing what they had been instructed. A summary of the results of the observations is included in the appendices. Four hypotheses, two major and two minor, were evaluated in the investigation.

Results

In the review of the literature in Chapter II it was not clear that investigators have been able to show conclusively that using laboratories or discussion sections in addition to lecture is more effective than lecture alone. There are individual studies that have been able to show increases in achievement and attitude (c.f. Bradley, 1986; Gunsch, 1972; Hofstein, Gluzman, Ben-Zvi, & Samual, 1980; Lawrenz & Munch, 1984; Saunders & Dickinson, 1979; White & Rosenthal, 1974), but when the studies of 1930 to 1980 were reviewed and the results summarized it was found that a very small percentage were able to show a significant difference.

Even though the existing body of research is unable to show conclusively that laboratory sessions increase achievement and attitude (c.f. Boghai, 1978; Grozier, 1969; Vance & Kieren, 1971; Yager, Engen, & Snider, 1969), laboratories are still recommended and used in science classes throughout the country. The 1991 position statement of the National Science Teachers Association (1991) stresses that laboratories should be a required component at all levels of science education.

The review of the computer science education literature shows that closed laboratories are just now being recommended and used in universities across the nation. In light of these recommendations and with the high cost of implementing closed laboratories the investigator attempted to examine the effect of closed laboratories on concept comprehension and perception of effectiveness of the course on students enrolled in a computer science course required of majors.

The first hypothesis to be tested dealt with the effectiveness of comprehension of five selected concepts. The adjusted mean scores for the treatment group were compared to those of the control group and no significant difference was found. The review of similar studies performed in different areas of science support this conclusion (c.f. Boghai, 1978; Grozier, 1969; Vance & Kieren, 1971; Yager, Engen, & Snider, 1969). The two methods examined in this investigation, closed laboratory activities and traditional discussion sections, both appear to effectively allow students to gain a better understanding of the abstract concepts covered in lecture. Specifically, one method allows students to discover the answers to problems by running a computer program designed for that purpose. The other method requires students to figure the answers with pencil and paper, and then discuss them with a Teaching Assistant. The nature of the course as taught at the major southwest research university used in this study supports these methods of supplementary instruction. There are two very large lecture sections, each with over 100 students, which do not lend themselves to giving individual attention to students. The discussion sections which are usually less than 30 students allow the Teaching Assistants to address the questions of students and the students, presumably, to be more comfortable asking them.

Because many studies of the effectiveness of laboratories have determined that students identified as low to middle achievers gain the most from the inclusion of laboratories, the first subhypothesis tested the effect of closed laboratory activities on the concept comprehension of students identified as less likely to succeed. There have been many studies devised to find the best predictors of success for students in computer science courses. Many of these predictors are for the initial course taken. All of the students registered for the course used in this study have already passed the first computer science course. Even though there was enough difference between the students to allow them to be identified as less likely to succeed or as more likely to succeed, the predictors used were not infallible in the present instance. Table 8 shows the distribution of students who passed the course and whether they were classified as less likely to succeed or not for all students who filled out the Burton Profile Questionnaire. 56% of those students identified as less likely to succeed passed the course.

Table 8

Likelihood to Succeed Versus Passed Course

Group	Passed Course	%Passed of Group	Failed Course	%Failed of Group
Less Likely	62	56	49	44
More Likely	72	65	38	35

The result of the statistical analysis for the differences in comprehension between the control and treatment groups of students identified as less likely to succeed was not significant so the null hypothesis was not rejected. There was no difference in concept comprehension by the students attending both closed laboratories and discussion sections and the students attending traditional discussion sections alone. One explanation for this is that the subgroup of students identified as less likely to succeed was not significantly different from the rest of the students and therefore a difference in concept comprehension for this subgroup would not be expected.

The two other hypotheses dealt with the results of students completing the Burton Perception Instrument. The fact that the date to administer course evaluations was listed in the syllabus and students knew that this was all that was going to be done that day may have resulted in a very high rate of absenteeism. The number of students actually used to test the hypotheses was extremely low leading to a very high chance of Type II error, meaning a better chance to fail to reject a false null hypothesis. When the means of the raw scores for the responses of the students are examined, they are very close leading this investigator to believe there was no significant difference, as the statistical analysis revealed. Results from previous studies dealing with closed laboratories in computer science have noted that the students like them and felt they made the course more effective. None of these previous studies have been experiments, only anecdotal findings from discussions with the students or from asking them specifically about their feelings for the exercises (c.f. Chavey, 1991; Penny & Ashton, 1990; Weiner, 1989; Wilson, 1991). This investigation was a carefully planned investigation designed to remove any and all threats to validity such as the Hawthorne Effect, control group rivalry, and the John Henry Effect. (Borg & Gall, 1989).

Conclusions

Based on the results described above, the investigator concludes that the use of closed laboratories is no more effective than traditional discussion sections when used to supplement computer science lecture sections. This is an important finding because of the expense involved in setting up and staffing closed laboratories. In order to have closed laboratory activities, computer rooms should be equipped with a computer for each student, or at most three students per computer, and the students should have access to them for the entire time specified for the closed laboratory session. The closed laboratory by definition requires that a laboratory instructor be present during the activities and that the room being used for the closed laboratory activities should be limited to this activity alone; in other words there should be no other students present doing other assignments at the same time. This causes problems to arise at many institutions such as not having enough facilities to keep other students from accessing the computer room when closed laboratory activities are taking place and/or not having the extra personnel necessary to staff them. The development of the closed laboratory activities themselves is very time consuming, and many instructors do not have extra time to devote to this purpose. The results of this investigation suggest that holding discussion sections in addition to lecture is just as effective as holding closed laboratory sessions. Unfortunately this does not help the very small colleges and universities that do not have graduate students or others to staff the discussion sections, but it does suggest that they do not have to feel pressured into developing programs, buying computers, or taking away existing computer access time from other students in order to hold closed laboratories.

Another area studied in the investigation was that of students perception of the effectiveness of the course. The statistical results showed no significant differences between the control and treatment groups. Yet when informally discussing the use of the closed laboratories with the Teaching Assistants who administered them they all felt that the closed laboratories were more effective than the traditional discussion section method. The conclusion drawn from these results is that the five weeks of closed laboratories was not enough to effect the students' overall perception of the course. Even when the perceptions of just the discussion sections/closed laboratories as measured by the Burton Perception Instrument are examined alone there is no significant difference.

Suggestions for Further Research

The present study was limited by two factors, small sample size and limited exposure to closed laboratories. The investigator suggests that the study should be replicated with a larger sample and with closed laboratories for every concept.

Even though this investigation supported the null hypotheses that there were as no significant differences in the effectiveness of closed laboratories and traditional discussion sections, there is a need for much more research to be done in this area. This investigation used only one type of closed laboratory activity. There are many other types and these too should be examined for their effectiveness (e.g. experimental, descriptive).

Another area that should be studied is whether the discussion sections and/or closed laboratories are necessary. The existing literature from science and mathematics is inconclusive. There are many small colleges and universities feeling pressured to include these supplementary activities when they do not have the personnel or resources to do so. Computer science education would benefit from research to determine conclusively whether these supplementary activities are necessary at all.

The present study was the first attempt to document the effectiveness of closed laboratories in a rigorous experiment. It sets the stage for further research on the effectiveness of closed laboratories and determining which types of closed laboratory activities are the most effective.

Appendix A

Consent Form

Consent Form

You are invited to participate in a study of the use of alternative learning environments. The effect, if any, of the alternative learning environments are going to be measured. There will be about 350 students from The University of Texas Computer Sciences Department in this study.

If you are willing to participate, **there are no specific tasks you must perform outside the normal requirements of your computer science courses.** Any information that is obtained from you or about you in connection with this study will remain confidential and will be disclosed only with your permission. We do not foresee any reason to disclose data from any individual: We are only interested in group data.

Your decision whether or not to participate will not prejudice your future relations with The University of Texas at Austin. If you decide to participate, you are free to discontinue participation at any time without prejudice.

If you have any questions, please ask now. If you have any additional questions later, Professor D. Burton, 471-9577, will be happy to answer them.

You will be offered a copy of this form to keep.

You are making a decision whether or not to participate in this important project. Your signature indicates that you have read the information provided above and have chosen to participate. You may withdraw at any time without prejudice after signing this form.

Signature ______________________ Date ____________

Project ID number ______________
(last 5 digits of social security number)

Please Print Your Name Here ______________________ Signature of Investigator ______________________

Appendix B

Burton Comprehension Instrument (Pretest)

CS 315 Preliminary Test

Please fill in the letter of the BEST answer to each of the following questions on the scantron.

1. Consider the following four tasks:
    1. To perform a linear search of a list of n names
    2. To perform a binary search of a sorted list of n names
    3. To perform a selection sort into alphabetical order of a list of n names that are initially in random order
    4. To perform a merge sort into alphabetical order of a list of n names that are initially in random order

For large n, which of the following lists these tasks in order (from least to greatest) of their worst-case running times?
(a) 2, 4, 3, 1 (b) 4, 3, 1, 2 (c) 2, 4, 1, 3
(d) 2, 1, 4, 3 (e) 2, 3, 1, 4

2. If N is the number of data elements to be manipulated by one or more algorithms, then the execution time of each such algorithm can be characterized in terms of N using the Big-O notation. Of the following, which best characterizes an execution time that is significantly different from the others?
(a) O(N) (b) O(2N) (c) O(N + 2)
(d) O(N/2) (e) O(N)

3. An ADT where the retrieval operation returns the item that has been in the structure the least amount of time.
(a) list (b) stack (c) FIFO queue
(d) priority queue (e) circle

4. An ADT where the retrieval operation returns the item that has been in the structure the longest time.
(a) list (b) stack (c) FIFO queue
(d) priority queue (e) circle

5. Of the following data representations for storing integers with an arbitrary number of digits, which would allow two integers having that representation to be added and the result to be stored using the same representation in the most time-efficient manner?
(a) A binary search tree with the nodes containing the digits and their positions in the decimal representation of the integer and with the nodes ordered by the digits
(b) A stack of digits, with the leftmost (most significant) digit on top of the stack
(c) A stack of digits, with the rightmost (least significant) digit on top of the stack
(d) A linked list of digits, proceeding from the leftmost (most significant) digit to the rightmost (least significant) digit
(e) A linked list of digits, proceeding from the rightmost (least significant) digit to the leftmost (most significant) digit

6. function z(k,n:integer):integer;

```
    begin
    if n = k then
        z := k
    else
        if n > k then
             z := z(k,n-k)
        else
             z := z(k-n,n)
    end;
```

Based on the function defined above, what is the value of z(6,8)?
(a) 1 (b) 2 (c) 3 (d) 4 (e) 8

7. It would be most appropriate to use a recursive function or procedure to solve a problem that
(a) can be reduced to two, or more, simpler or smaller cases of the same problem
(b) involves a substantial number of conditionals and nested loops
(c) requires a lot of memory
(d) involves storing data in a two-dimensional array
(e) involves evaluation of the factorial function

8.
```
program Main;
    var z:integer;
    function F(x:integer):integer;
    begin
         if (x = 1) or (x = 3) then
              F := x
         else
              F := x * F(x - 1)
    end;
begin  (* main *)
z := F(F(2) + F(5))
end.
```

If MAXINT were large enough to allow the program above to be executed, then at the end of the program, the value of z would be
(a) 62 (b) 5! + 2! (c) (5! + 2!)!
(d) (7!)! (e) (62!)/(2!)

9. Merge sort (internal) has which of the following advantages over bubble sort for long lists?

i. Merge sort requires much less coding to implement than does bubble sort
ii. Merge sort runs faster than bubble sort
iii. Merge sort requires less storage space than bubble sort

(a) i only (b) ii only (c) iii only
(d) i and iii (e) ii and iii

10. How many comparisons are required to sort an array of length 5 if a straight selection sort is used?
(a) 5 (b) 10 (c) 15 (d) 20 (e) 25

11\. How many comparisons are required to sort an array of length 5 if a straight selection sort is used and the array is already sorted?
(a) 0 (b) 1 (c) 10 (d) 20 (e) 30

12\. How many comparisons are required to sort an array of length 5 if the bubble sort with a Boolean flag, ShortBubble, is used?
(a) 5 (b) 10 (c) 15 (d) 20 (e) 25

13\. How many comparisons are required to sort an array of length 5 if ShortBubble is used and the array is already sorted?
(a) 0 (b) 1 (c) 4 (d) 5 (e) 10

14\. In QuickSort, if the splitting value is the first element in the array, then QuickSort will be most efficient with input data in:
(a) sorted order (b) random order (c) nearly sorted order
(d) reverse order (e) law and order

Questions 15 and 16 deal with the following problem. The height of a binary tree is the number of nodes in the longest path from the root to a leaf of the tree. The height of an empty tree is 0; the height of a single-node tree is 1.

```
1   Function Height(Tree:TreeType):integer;
2   Begin
3     If ________________  then   (* question 15 *)
4         Height := 0
5     else
6       ____________________      (* question 16 *)
7   end;
```

15\. Fill the blank in line 3.
(a) Tree <> NIL (b) Tree = NIL (c) Tree^ = NIL
(d) Tree^.next = NIL (e) none

16\. Fill the blank in line 6.
(a) Height(Tree^.right) + Height(Tree^.left)
(b) Height(Tree^.right) + Height(Tree^.left) + 1
(c) Height := Height(Tree^.right) + Height(Tree^.left) + 1
(d) Height := Max(Height(Tree^.right), Height(Tree^.left)) + 1
(e) none

Questions 17 - 22 deal with the following problem.
You have a situation where you need to use two stacks, S1 and S2. You know that together they will never have more than Max elements. You decide to use an array representation with both stacks residing in the same array but with two logical pointers to top, Top1 (for S1) and Top2 (for S2). Notice this will not be a linked list. Use the following definitions and declarations.

```
CONST
    Max = 100;
    MaxPlus1 = 101;
TYPE
    ItemType = (* component on the stack *)
    StackType = record
                   Stack:array [1..Max] of ItemType;
                   Top1: 0 .. Max;
                   Top2: 1 .. MaxPlus1
                end;
```

17. Choose the statement that correctly implements this procedure.

```
Procedure ClearS2(Var S2:StackType);
Begin
    (* statement goes here *)
end;
```

a. S2.Top2 := 0;
b. S1.Top1 := 0;
c. S2.Top2 := MaxPlus1;
d. S2.Top2 := Max;
e. none

18. Choose the statement that correctly implements this function.

```
Function Full(S1,S2:StackType):boolean;
begin
    (* statement goes here *)
end;
```

a. Full := S1.Top1 = Max;
b. Full := S1.Top1 = S2.Top2;
c. Full := S2.Top2 - S1.Top1 = 1;
d. Full := S1.Top1 + S2.Top2 = MaxPlus1;
e. none

Questions 19 and 20 deal with the following procedure shell.

```
Procedure PushS1(Var S1:StackType; Item:ItemType);
Begin
    S1.Top1 := ____________________; (* question 19 *)
    ____________________ := Item  (* question 20 *)
end;
```

19. Choose the expression that makes the first statement correct.
(a) S1.Top1 + 1   (b) S1.Top1 - 1   (c) Max - S1.Top1
(d) S1.Top1 + MaxPlus1   (e) none

20. Choose the expression that makes the second statement correct.
(a) Stack[S1.Top1 + 1]   (b) Stack[MaxPlus1 - S1.Top1]
(c) Stack[S1.Top1]   (d) S1.Stack[S1.Top1]   (e) none

Questions 21 and 22 deal with the following procedure shell.

```
Procedure PopS2(Var S2:StackType; Var Item:ItemType);
Begin
    Item := _______________;  (* question 21 *)
    S2.Top2 := ____________;  (* question 22 *)
end;
```

21. Choose the expression that makes the first statement correct.
(a) Stack[S2.Top2] (b) Stack[Max - S2.Top2]
(c) S2.Stack[Top2] (d) S2.Stack[S2.Top2] (e) none

22. Choose the expression that makes the second statement correct.
(a) S2.Top2 + 1 (b) S2.Top2 - 1
(c) MaxPlus1 - Top2 (d) Max - S2.Top2 (e) none

Questions 23 through 25 use the following table.

N	column a	column b	column c
32	5	1024	160
64	6	4096	384
128	7	16384	896
256	8	65536	2048
512	9	262144	4608
1024	10	1048576	10240
2048	11	4194304	22528
4096	12	16777216	49152

You have run different algorithms to accomplish the same task and these were the values that were returned as *amount of work* for the appropriate value of N (first column).

23. Which of the following would most closely describe the complexity of the algorithm returning the values in column a?

(a) N (b) logN (c) NlogN (d) N^2 (e) 2^N

24. Which of the following would most closely describe the complexity of the algorithm returning the values in column b?

(a) N (b) logN (c) NlogN (d) N^2 (e) 2^N

25. Which of the following would most closely describe the complexity of the algorithm returning the values in column c?

(a) N (b) logN (c) NlogN (d) N^2 (e) 2^N

Appendix C

Reliability Measures for Pretest

Coefficents of Correlation
for the Items on the Pretest

Item Number	Coef. of Corr. Measure
1	0.47
2	0.27
3	0.42
4	0.38
5	0.51
6	0.37
7	0.38
8	0.60
9	0.39
10	0.42
11	0.51
12	0.50
13	0.53
14	0.42
15	0.41
16	0.52
17	0.30
18	0.23
19	0.38
20	0.55
21	-0.08 *
22	0.45
23	0.34
24	0.24
25	0.25

* Questions #21 was left in because it was one-half of a two part question which only one student missed on the semester of the pilot test and that student was in the top quartile of student grades resulting in a low reliability measure.

Appendix D

Burton Profile Questionnaire

Confidential Questionnaire

Please answer each of the following by filling in the appropriate circle on the scantron answer sheet.

1. Fill in your Project ID number (last five digits of your social security number) in the name field of the scantron form.

2. Your gender:

a. female b. male

3. Your ethnicity:

a. white b. asian c. other

4. Your classification:

a. freshaman b. sophomore c. junior d. senior e. graduate

5. Approximate number of hours you will be working per week:

a. zero b. 1 to 19 c. 20 or more

6. Number of computer courses in high school:

a. zero b. 1 c. 2 d. 3 or more

7. Number of mathematics courses in high school (past Algebra I):

a. 1 b. 2 c. 3 d. 4 or more

8. Number of computer courses in college:

a. 0 b. 1 c. 2 or more

9. Number of mathematics courses in college:

a. 0 b. 1 c. 2 or more

10. Approximate number of years of computer experience (school and/or other):

a. 0 b. 1/2 c. 1 or more

11. Approximate G.P.A. in high school:

a. A b. B c. C d. other

12. Approximate G.P.A. in college:

a. A b. B c. C d. none e. other

13. Have you taken CS 315 before?

    a. yes   b. no

14. Did you place out of CS 304P?

    a. yes   b. no

For the following questions, if you cannot remember or cannot make a reasonable estimation, leave it blank.

15. Approximate ACT English score (if took ACT):

    a. greater than or equal to 30
    b. greater than or equal to 24 and less than 30
    c. greater than or equal to 21 and less than 24
    d. greater than or equal to 18 and less than 21
    d. less than 18

16. Approximate ACT Math score (if took ACT):

    a. greater than or equal to 30
    b. greater than or equal to 24 and less than 30
    c. greater than or equal to 21 and less than 24
    d. greater than or equal to 18 and less than 21
    d. less than 18

17. Approximate ACT composite score (if took ACT):

    a. greater than or equal to 30
    b. greater than or equal to 24 and less than 30
    c. greater than or equal to 21 and less than 24
    d. greater than or equal to 18 and less than 21
    d. less than 18

18. Approximate SAT Verbal score (if took SAT):

    a. greater than or equal to 550
    b. greater than or equal to 475 and less than 550
    c. greater than or equal to 400 and less than 475
    d. less than 400

19. Approximate SAT Math score (if took SAT):

    a. greater than or equal to 550
    b. greater than or equal to 475 and less than 550
    c. greater than or equal to 400 and less than 475
    d. less than 400

20. Approximate SAT total score (if took SAT):

    a. greater than or equal to 1100
    b. greater than or equal to 950 and less than 1100
    c. greater than or equal to 800 and less than 950
    d. less than 800

Appendix E

Table of ACT/SAT Scores from The University of Texas at Austin

The University of Texas at Austin
Measurement and Evaluation Center
Box 7246, Austin, Tx 78713-7246
(512) 471-3032

ESTIMATED PROBABILITY OF C OR BETTER GPA AT END OF FRESHMAN YEAR AT UT AUSTIN

September, 1991

This information is intended to help counselors discuss with students their chances of doing well at UT Austin.

For this purpose, "doing well" means earning a grade point average (GPA) of C or better by the end of the freshman year. The data in the tables represent the statistical chances of success. Of course, such predictions tell only part of the story. In predicting academic success for the individual student, many factors must be considered in addition to test scores and high school performance.

The probability estimates in Table 1 below are based on the academic performance of 5207 freshmen who entered UT Austin in the fall of 1990. These data were obtained by relating the students' College Board Scholastic Aptitude Test (SAT) Total scores and their high school class percentile ranks to their UT Austin GPA's at the end of the Spring Semester, 1991.

SAT-Total scores (Verbal plus Mathematical) are shown in the column at the left; percentile ranks in high school class are given across the top of the table. To find the chance of success, note that a student with an SAT-Total score of 1000 and a high school percentile rank of 85 has a probability estimate of 77. This means that approximately 77 out of every 100 freshmen in 1990–1991 who had SAT-Total scores of 1000 and high school ranks of 85 made a C average or better during their freshman year at UT Austin. [Conversely, approximately 23 (100 minus 77 students with SAT-Total scores of 1000 and high school ranks of 85) made less than a C average.]

Table 1. Probabilities Based on SAT Data from 1990–1991 Academic Year

ALL UNIVERSITY
Percentile Rank in High School Class

SAT TOTAL SCORE	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	SAT TOTAL SCORE
1600	50	55	61	66	71	75	80	83	87	89	92	94	95	96	97	98	99	99	99	1600
1550	45	50	56	61	66	71	76	80	84	87	90	92	94	95	97	97	98	99	99	1550
1500	40	45	51	56	62	67	72	76	80	84	87	90	92	94	95	97	98	98	99	1500
1450		40	46	51	57	62	68	72	77	81	84	87	90	92	94	96	97	98	98	1450
1400			41	47	52	58	63	68	73	77	81	85	88	90	93	94	96	97	98	1400
S 1350				42	47	53	58	63	69	73	78	82	85	88	91	93	94	96	97	1350 S
A 1300					42	48	53	59	64	69	74	78	82	85	88	91	93	95	96	1300 A
T 1250						43	48	54	59	65	70	74	79	82	86	89	91	93	95	1250 T
1200							43	49	54	60	65	70	75	79	83	86	89	91	93	1200
T 1150								44	49	55	60	66	71	75	79	83	86	89	92	1150 T
O 1100									44	50	56	61	66	71	76	80	83	87	89	1100 O
T 1050										45	51	56	61	67	72	76	80	84	87	1050 T
A 1000										40	46	51	57	62	67	72	77	81	84	1000 A
L 950											41	46	52	57	63	68	73	77	81	950 L
900												41	47	52	58	63	68	73	77	900
S 850															53	58	64	69	74	850 S
C 800															48	53	59	64	69	800 C
O 750																		59	65	750 O
R 700																		55	60	700 R
E 650																		50	55	650 E
600																		45	50	600
550																		40	45	550
500																			40	500
450																				450
400																				400

LESS THAN 40

NOT ELIGIBLE FOR REGULAR ADMISSION

Prepared by Measurement and Evaluation Center
The University of Texas at Austin

The probability estimates in Table 2 below are based on the academic performance of 978 freshmen who entered U. T. Austin in the fall of 1990 and who took the ACT after September,1989. These data were obtained by relating the students' American College Testing (ACT) Assessment Composite scores and their high school class percentile ranks to their U.T. Austin GPA's at the end of the Spring Semester, 1991.

Table 2. Probabilities Based on ACT Data from 1990–1991 Academic Year

ALL UNIVERSITY
Percentile Rank in High School Class

ACT COMPOSITE SCORE	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	
35	40	45	50	55	60	65	69	73	77	81	84	87	89	91	93	95	96	97	98	35
34		42	47	52	57	61	66	70	75	78	82	85	88	90	92	94	95	96	97	34
33			43	48	53	58	63	67	72	76	80	83	86	88	91	93	94	95	97	33
32			40	45	50	55	60	64	69	73	77	81	84	87	89	91	93	95	96	32
31				42	46	51	56	61	66	70	74	78	82	85	88	90	92	94	95	31
30					43	48	53	58	63	67	72	76	79	83	86	88	91	93	94	30
29					40	45	50	55	59	64	69	73	77	81	84	87	89	91	93	29
28						41	46	51	56	61	66	70	74	78	82	85	87	90	92	28
27							43	48	53	58	63	67	71	76	79	83	86	88	91	27
26							40	45	49	54	59	64	69	73	77	80	84	87	89	26
25										51	56	61	66	70	74	78	81	85	87	25
24										48	53	58	62	67	71	75	79	83	86	24
23										44	49	54	59	64	68	73	77	80	84	23
22										41	46	51	56	61	65	70	74	78	81	22
21											43	48	53	57	62	67	71	75	79	21
20															59	64	68	73	77	20
19															56	61	65	70	74	19
18																		67	71	18
17																		64	68	17
16																		60	65	16
15																		57	62	15
14																		54	59	14
13																		50	55	13
12																		47	52	12
11																		44	49	11
10																		40	45	10
9																			42	9
8																				8
7																				7
6																				6
5																				5
4																				4
3																				3
2																				2
1																				1

LESS THAN 40

NOT ELIGIBLE FOR REGULAR ADMISSION

Prepared by the Measurement and Evaluation Center
The University of Texas at Austin

Appendix F

Closed Laboratory and Discussion Section Activities

Name: ____________________ TA's Name: ____________________

Day: ____________ Time: ________________

Testing a Program

The objective of this exercise is to write and implement a test plan for a given program. The disk contains an executable program, you only need to type the command 'tester' at the prompt.

The program has the following specification: (1) the input is three positive integers that each represent the side of a triangle, (2) the output of the program is one of the following terms which identifies the type of triangle represented by the three values that were input: equilateral (all three sides are equal); isosceles (two of the three sides are equal); or scalene (none of the three sides are equal).

First you should write a test plan designed to find any possible errors in the program. Next you should verify your test plan by running the program on the disk. The program has several versions of the triangle program each of which contain one or no errors. You should apply your test plan to one version at a time until you either find the error or are convinced it has no error and then go onto the next version.

You may use the following test plan outline to help you. Please be sure and identify what you are testing (test case) with each input. (you may have more or less test cases than there are lines, this guide is not supposed to be representative of the number of cases in the test plan)

Test case	input	expected output	actual output (for each version)						
			1	2	3	4	5	6	7

Test case	input	expected output	actual output (for each version)						
			1	2	3	4	5	6	7

Name: ______________________ TA's Name: __________________

Day: ____________ Time: ________________

Testing a Program

The objective of this exercise is to write a test plan for a given program.

The program has the following specification: (1) the input is three positive integers that each represent the side of a triangle, (2) the output of the program is one of the following terms which identifies the type of triangle represented by the three values that were input: equilateral (all three sides are equal); isosceles (two of the three sides are equal); or scalene (none of the three sides are equal).

You may use the following test plan outline to help you. Please be sure and identify what you are testing (test case) with each input. (you may have more or less test cases than there are lines, this guide is not supposed to be representative of the number of cases in the test plan)

Test case	input	expected output	actual output

Test case	input	expected output	actual output

Relative Complexity of Algorithms

This lab is intended to help you understand the relative complexity of two algorithms. You will run a program that allows you to execute pairs of algorithms performing the same task. Each of these algorithms is of a different class of complexity. You are to approximate the Big-O values for each algorithm, determine which one (A or B) would be better for large values of N, and determine at what value of N (size of input) one algorithm surpasses the other.

You will examine four pairs of algorithms. Please answer the following questions about each pair. You were given a disk which has the program on it, please return it at the end of the class period.

To run the program type bigo at the prompt.

Program Limits:

--maximum array size that can be entered is 10,000

--values to be searched for in arrays must be positive integers between 1 and twice the length of the array being searched

--when you select 0 to stop, a summary of your trial results are printed to the screen, you may enter as many trials as you wish to determine the Big-O values, but if you use more than 20 trials only the last 20 will be printed in the summary

---

Answer the following questions in the order in which they are encountered.

I. Calculate the sum of the elements in an array.

1. Please repeatedly enter different values of N (size of array to be summed) and record the amount of work below. After you see a pattern developing as the size of N increases, identify the appropriate Big-O values for these two algorithms. You do not have to record every entry, just ones that show the trend that allowed you to choose the Big-O values that you did.
(HINT: start with small values of N)

N	A	B

2. Based on these results what are the Big-O values for algorithms A and B?

A = ____________ B = ____________

3. Based on these estimations which algorithm (A or B) would you choose to use for this application (with very large N)? ________

4. At what value of N does your choice in #2 become better than the other? _____

II. Searching for values in an ordered array.

1. Please repeatedly enter different values of N (size of array to be searched) and different values to be searched for (between 1 and twice the size of the array) then record the amount of work below. After you see a pattern developing as you try different sizes of N, identify the appropriate Big-O values for these two algorithms. You do not have to record every entry, just ones that show the trend that allowed you to choose the Big-O values that you did. (HINT: start with large values of N)

N	Search Value	A	B
______	______	______	______
	______	______	______
	______	______	______
	______	______	______
	______	______	______
______	______	______	______
	______	______	______
	______	______	______
	______	______	______
	______	______	______
______	______	______	______
	______	______	______
	______	______	______
	______	______	______
	______	______	______

2. Based on these results what are the Big-O values for algorithms A and B?

A = ______________ B = ______________

3. Based on these estimations which algorithm (A or B) would you choose to use for this application? _________

4. At what value of N does your choice in #2 become better than the other? _____

III. Sorting an unordered array.

1. Please repeatedly enter different values of N (size of array to be sorted) and record the amount of work below. After you see a pattern developing as the size of N increases, identify the appropriate Big-O values for these two algorithms. You do not have to record every entry, just ones that show the trend that allowed you to choose the Big-O values that you did.

N	A	B

2. Based on these results what are the Big-O values for algorithms A and B?

A = ______________ B = ______________

3. Based on these estimations which algorithm (A or B) would you choose to use for this application? _________

4. At what value of N does your choice in #2 become better than the other? _____

IV. Calculate the values of 2^N.

1. Please repeatedly enter different values of N (power of two desired) and record the amount of work below. After you see a pattern developing as the size of N increases, identify the appropriate Big-O values for these two algorithms. You do not have to record every entry, just ones that show the trend that allowed you to choose the Big-O values that you did. (HINT: start with N equal to 1 and increment by 1)

N	A	B

2. Based on these results what are the Big-O values for algorithms A and B?

A = ______________ B = ______________

3. Based on these estimations which algorithm (A or B) would you choose to use for this application? _________

4. At what value of N does your choice in #2 become better than the other? _____

Relative Complexity of Algorithms

1. Please place a number (1 to 5) beside each of the following complexity classes to signify the relationship between the classes, 1 is best to 5 is worst.

_____ O(N^2)

_____ O(N log N)

_____ O(log N)

_____ O(N)

_____ O(1)

2. Answer the following questions which refer to the algorithms below:

```
 PROCEDURE Sum1 (n : integer);
    VAR sum : integer;
    BEGIN
      sum := (n * (n + 1)) DIV 2
    END;

  PROCEDURE Sum2 (n : integer);
    VAR i, sum : integer;
    BEGIN
      sum := 0;
      FOR i := 1 TO n DO
        sum := sum + i;
    END;
```

a. Approximate the Big-O values of each of these algorithms:

Big-O of Sum1: _________

Big-O of Sum2: ________

b. Based on these values of Big-O which of these algorithms would you choose for very large N? ______

c. At approximately what value of N does your choice become better than the algorithm you did not choose? _____

3. Answer the following questions which refer to the algorithms below:

```
PROCEDURE Search1
    (A : ArrayType; KeyVal : Integer; VAR Loc : Integer);
VAR MoreToSearch : Boolean;
BEGIN
  Loc := 1;
  MoreToSearch := True;
  WHILE MoreToSearch AND (Loc <= A.Length) DO
      IF A.Data[Loc] < KeyVal THEN
         Loc := Loc + 1
      ELSE
         MoreToSearch := False;
  IF Loc > A.Length THEN
      Loc := 0
  ELSE
      IF A.Data[Loc] <> KeyVal THEN
         Loc := 0
END;

PROCEDURE Search2
            (A : ArrayType; KeyVal : Integer; VAR Loc : Integer);
VAR Found : Boolean;
    First, MidPoint, Last : ArrayRange;
BEGIN
  Found := False;
  First := 1;
  Last := A.Length;
  While (First <= Last) AND NOT Found DO
      Begin
        MidPoint := (First + Last) DIV 2;
        IF A.Data[MidPoint] = KeyVal THEN
           Found := True
        ELSE
           IF A.Data[MidPoint] > KeyVal THEN
              Last := MidPoint - 1
           ELSE
              First := MidPoint + 1
      end;
  IF Found THEN
     Loc := MidPoint
  ELSE
     Loc := 0
END;
```

a. Approximate the Big-O values of each of these algorithms:

Big-O of Search1: _________

Big-O of Search2: ________

b. Based on these values of Big-O which of these algorithms would you choose for very large N? ______

c. At approximately what value of N does your choice become better than the algorithm you did not choose? ______

4. a. If you were given two algorithms to perform the same task and were told that one of the algorithms had Big-O of N2 and the other algorithms had Big-O of 2N, which of these algorithms would you choose to use if you new your N would be very large? ______

b. At approximately what value of N would your choice become better than the algorithm you did not choose? ______

5. Answer in your own words. What is the Big-O value of an algorithm and what is it used for?

Name: ______________ TA Name: ________________

Day: ________ Time: ________

Stacks and Queues

The objective of these exercises is to allow you to become proficient in the basic operations performed on stacks and FIFO queues.

For each of the following questions: (1) work out the answer by hand simulating the stack or queue operations as specified, (2) verify your answer by using the 'stack' or 'queue' programs.

To run the programs you need only enter the command 'stack' or 'queue' at the prompt. By typing HELP the programs will instruct you on the proper method of entering commands. If you have any questions or difficulties feel free to ask a neighbor or your TA.

I. Stacks.

1. After the following commands are executed what would be the next item on the top of the stack.

```
PUSH A
PUSH S
PUSH T
POP
POP
PUSH R
PUSH O
POP
POP
```

2. After the following commands are executed what would be the next item on the top of the stack.

```
PUSH O
PUSH I
POP
PUSH L
PUSH E
POP
PUSH R
POP
POP
```

3. The following code segment reads five data values and either prints them or puts them on a stack. After all of the values are read, the items on the stack are popped and printed. Because of the nature of a stack, this code segment cannot print certain combinations of the input data. RanFun is a BOOLEAN function that determines whether a value is to be written or put on the stack. Its actual behavior is immaterial.

```
FOR Count := 1 to 5 DO
     BEGIN
       READ(ch);
       IF RanFun THEN
          WRITE(ch)
       ELSE
          Push(S,ch)
     END;
WHILE NOT Empty(S) DO
     BEGIN
       Pop(S,ch);
       WRITE(ch)
     END;
```

If the input values are A, B, C, D, and E (in that order), please identify each of the following as (P) possible or (I) impossible outputs of this code. Use the 'stack' program to verify your results.

A B C D E ______________

A C E B D ______________

A B E B C ______________

E D C B A ______________

E C D A B ______________

A B E D C ______________

II. Queues.

1. After the following commands are executed what would be the next item at the front of the queue.

```
ENQUEUE T
ENQUEUE E
ENQUEUE X
DEQUEUE
DEQUEUE
ENQUEUE A
ENQUEUE S
DEQUEUE
DEQUEUE
```

2. After the following commands are executed what would be the next item at the front of the queue.

```
ENQUEUE R
ENQUEUE I
DEQUEUE
ENQUEUE D
ENQUEUE E
DEQUEUE
ENQUEUE R
DEQUEUE
DEQUEUE
```

3. The following code segment reads five data values and either prints them or puts them in a FIFO queue. After all of the values are read, the items in the queue are dequeued and printed. Because of the nature of a FIFO queue, this code segment cannot print certain combinations of the input data. RanFun is a BOOLEAN function that determines whether a value is to be written or put in the queue. Its actual behavior is immaterial.

```
FOR Count := 1 to 5 DO
     BEGIN
       READ(ch);
       IF RanFun THEN
          WRITE(ch)
       ELSE
          Enqueue(Q,ch)
     END;
WHILE NOT Empty(Q) DO
     BEGIN
       Dequeue(Q,ch);
       WRITE(ch)
     END;
```

If the input values are A, B, C, D, and E (in that order), please identify each of the following as (P) possible or (I) impossible outputs of this code. Use the 'queue' program to verify your results.

A B C D E ______________

A C E B D ______________

A B E B C ______________

E D C B A ______________

E C D A B ______________

A B E D C ______________

Name: ______________ TA Name: ________________

Day: ________ Time: ________

Stacks and Queues

The objective of these exercises is to allow you to become proficient in the basic operations performed on stacks and FIFO queues.

I. Stacks.

1. After the following commands are executed what would be the next item on the top of the stack.

```
PUSH A
PUSH S
PUSH T
POP
POP
PUSH R
PUSH O
POP
POP
```

2. After the following commands are executed what would be the next item on the top of the stack.

```
PUSH O
PUSH I
POP
PUSH L
PUSH E
POP
PUSH R
POP
POP
```

3. The following code segment reads five data values and either prints them or puts them on a stack. After all of the values are read, the items on the stack are popped and printed. Because of the nature of a stack, this code segment cannot print certain combinations of the input data. RanFun is a BOOLEAN function that determines whether a value is to be written or put on the stack. Its actual behavior is immaterial.

```
FOR Count := 1 to 5 DO
     BEGIN
       READ(ch);
       IF RanFun THEN
          WRITE(ch)
       ELSE
          Push(S,ch)
     END;
```

```
          WHILE NOT Empty(S) DO
                BEGIN
                  Pop(S,ch);
                  WRITE(ch)
                END;
```

If the input values are A, B, C, D, and E (in that order), please identify each of the following as (P) possible or (I) impossible outputs of this code.

A B C D E ______________

A C E B D ______________

A B E B C ______________

E D C B A ______________

E C D A B ______________

A B E D C ______________

II. Queues.

1. After the following commands are executed what would be the next item at the front of the queue.

```
ENQUEUE T
ENQUEUE E
ENQUEUE X
DEQUEUE
DEQUEUE
ENQUEUE A
ENQUEUE S
DEQUEUE
DEQUEUE
```

2. After the following commands are executed what would be the next item at the front of the queue.

```
ENQUEUE R
ENQUEUE I
DEQUEUE
ENQUEUE D
ENQUEUE E
DEQUEUE
ENQUEUE R
DEQUEUE
DEQUEUE
```

3. The following code segment reads five data values and either prints them or puts them in a FIFO queue. After all of the values are read, the items in the queue are dequeued and printed. Because of the nature of a FIFO queue, this code segment cannot print certain combinations of the input data. RanFun is a BOOLEAN function that determines whether a value is to be written or put in the queue. Its actual behavior is immaterial.

```
FOR Count := 1 to 5 DO
     BEGIN
       READ(ch);
       IF RanFun THEN
          WRITE(ch)
       ELSE
          Enqueue(Q,ch)
     END;
WHILE NOT Empty(Q) DO
     BEGIN
       Dequeue(Q,ch);
       WRITE(ch)
     END;
```

If the input values are A, B, C, D, and E (in that order), please identify each of the following as (P) possible or (I) impossible outputs of this code.

A B C D E ______________

A C E B D ______________

A B E B C ______________

E D C B A ______________

E C D A B ______________

A B E D C ______________

Recursion

Answer each of the following questions. After you have answered a question you may run a program called 'recrsion' to verify your results.

When you run the program first select the E, for explanation choice, from the menu. This will explain how the interactive environment works for this program.

I.

```
    Function BinS(List:ListType; N, F, L:integer):integer;
    Var  M:integer;
    begin
         M := (F + L) DIV 2;
         If F > L then
              BinS := 0         (* returns zero if not found *)
         else
              If List[M] = N then
                   BinS := M    (* returns position if found *)
              else
                   If List[M] > N then
                        BinS := BinS(List, N, F, M - 1)
                   else
                        BinS := BinS(List, N, M + 1, L)
    end;
```

If the value of L is 10 and List contain the following:

8 11 24 26 30 37 59 74 97 97

Show the contents of the run-time stack if N (the value being searched for) is 25.

List and N are constant and do not have to be included as columns in your activation records.

F L BinS

The value of BinS returned to the main program is ________.

Show the contents of the runtime stack if the value being searched for is 8.

F L BinS

The value of BinS returned to the main program is _______.

II.

```
Function GCD(M, N:integer):integer;
begin
     If (N <= M) and (M mod N = 0) then
          GCD = N
     else
          if M < N then
               GCD := GCD(N, M)
          else
               GCD := GCD(N, M mod N)
end;
```

Show the contents of the run-time stack after the function has been called with M = 33 and N = 999.

M N GCD

The value of GCD returned to the main program is __________.

Name: ____________________

III. Run the other recursive routines if you have time, especially Factorial and Towers of Hanoi, to trace values through the run-time stack and watch how they are executed. When you are more comfortable with how the recursion is working write a paragraph explaining how and why recursion works in your own words. You may use one of the above functions (or any other) as an example in your paragraph if necessary.

Recursion

Answer each of the following questions.

I.

```
Function BinS(List:ListType; N, F, L:integer):integer;
Var  M:integer;
begin
    M := (F + L) DIV 2;
    If F > L then
         BinS := 0          (* returns zero if not found *)
    else
         If List[M] = N then
              BinS := M     (* returns position if found *)
         else
              If List[M] > N then
                   BinS := BinS(List, N, F, M - 1)
              else
                   BinS := BinS(List, N, M + 1, L)
end;
```

If the value of L is 10 and List contain the following:

8 11 24 26 30 37 59 74 97 97

Show the contents of the run-time stack if N (the value being searched for) is 25.

List and N are constant and do not have to be included as columns in your activation records.

F L BinS

The value of BinS returned to the main program is _________.

Show the contents of the runtime stack if the value being searched for is 8.

F	L	BinS

The value of BinS returned to the main program is _______.

II.

```
    Function GCD(M, N:integer):integer;
    begin
         If (N <= M) and (M mod N = 0) then
              GCD = N
         else
              if M < N then
                   GCD := GCD(N, M)
              else
                   GCD := GCD(N, M mod N)
    end;
```

Show the contents of the run-time stack after the function has been called with M = 33 and N = 999.

M	N	GCD

The value of GCD returned to the main program is __________.

Name: ____________________

III. Write a paragraph explaining how and why recursion works in your own words. You may use one of the above functions (or any other) as an example in your paragraph if necessary.

Sorting

Fill in the chart below as indicated for a list of 20 elements. You may run a program to assist you in your answers by typing "sort" at the prompt. You are to fill in the number of exchanges and the number of compares, the number of compares refers to actual comparison of data values not to loop control tests.

Ordered means the data is already sorted, reverse means that the data is sorted in reverse order, and scrambled means the data is random without any duplicates.

		Ordered	Reverse	Scrambled
Bubble Sort	Exchanges			
	Compares			
Bubble with flag	Exchanges			
	Compares			
Selection Sort	Exchanges			
	Compares			
Insertion Sort	Exchanges			
	Compares			
Quick Sort	Exchanges			
	Compares			
Heap Sort	Exchanges			
	Compares			

Does it matter which sort you use if you know the data is already sorted or is in reverse order?

Sorting

We have discussed sorting algorithms in class, you have seen the movie, and the algorithms are in the book. Based on these experiences please fill in the chart below. The examples that have been covered in class have used randomly ordered data, the following chart deals with the same algorithms being applied to data that is already sorted or is sorted in reverse order. Compares refers to actual comparisons of data values only, it does not include loop control tests.

You should place the word SAME, MORE, or LESS in each of the squares below. They should reflect the relation of the number of compares and exchanges for the data order indicated to the number of compares and exchanges with random data.

		Ordered	Reverse
Bubble Sort	Exchanges		
	Compares		
Bubble with flag	Exchanges		
	Compares		
Selection Sort	Exchanges		
	Compares		
Insertion Sort	Exchanges		
	Compares		
Quick Sort	Exchanges		
	Compares		
Heap Sort	Exchanges		
	Compares		

Does it matter which sort you use if you know the data is already sorted or is in reverse order?

Appendix G

Observation Instrument

TA Name: ________________ Day: _____ Time: _______

Observation Instrument

Please use the following scale.

1 - strongly disagree
2 - disagree
3 - neutral
4 - agree
5 - strongly agree

1. The TA explains the exercise to be performed

1 2 3 4 5

2. The TA asks for student questions

1 2 3 4 5

3. The TA answers questions satisfactorily (the students seem satisfied with the answers they receive from the TA)

1 2 3 4 5

4. The TA shows enthusiasm toward the exercise

1 2 3 4 5

5. The TA is responsive to the students

1 2 3 4 5

Please use the space below to make any comments necessary in interpreting the ratings given above.

Appendix H

Summary of Observations

Summary of Results of Observations

	TA Number	Average Rating
Ratings for First Observation:		
	0	3.2
	1	4.8
	2	2.8
	3c	3.2
	3d	4.0
	4	1.8
Ratings for Second Observation:		
	0	4.0
	1	3.6
	2	3.4
	3c	3.4
	3d	4.4
	4	2.2

The c and d signify the closed laboratory and discussion section group.

Appendix I

Burton Perception Instrument

Supplemental Course Evaluation

This questionnaire is to be completed in addition to your confidential course-instructor survey. The course-instructor survey is confidential and the results will not be known to the instructor until after grades have been issued. This additional questionnaire is part of the research study you have been participating in this semester.

Please mark your responses on the separate answer sheet. Be sure to use the correct question number. There are two sections of this questionnaire the first relates to the lecture portion of the course and the second relates to the discussion sections (including closed laboratory exercises).

Please remember to code your social security number (or student ID number) on the answer sheet.

Use the following response scale to answer the questions.

a = strongly agree
b = agree
c = neutral
d = disagree
e = strongly disagree

Section I. (this section refers to the lecture portion of the class alone)

1. The assignments contributed to the value of the course.

2. The assignments contributed to my learning.

3. The assignments were related to the course content.

4. The assignments helped prepare students for the exams.

5. The assignments were challenging.

6. The assignments were interesting.

7. Enough time was usually allowed to complete each assignment.

8. The examinations in this course were generally fair.

9. The examination questions corresponded to what had been taught.

10. The examinations permitted students to show understanding.

11. The examinations accurately measured my achievement in this course.

12. The examinations were graded objectively.

13. Enough time was allotted to attempt all the items on the examinations.

14. I am satisfied with the way the performance of students was evaluated in this course.

15. The overall organization of the course facilitated learning.

16. I was stimulated to work beyond the requirements of the course.

Section II. (this section refers to the discussion section/closed laboratory portion of the course)

17. The exercises contributed to the value of the course.

18. The exercises contributed to my learning.

19. The exercises were related to the course content.

20. The exercises helped prepare students for the exams.

21. The exercises were challenging.

22. The exercises were interesting.

23. Enough time was usually allowed to complete each exercise.

24. The exercises were fun.

25. The exercises helped prepare students for the programming assignments.

Appendix J

Alpha Coefficient Values for the Items

Cronbach's Alpha Coefficient Values

for Items on the Burton Perception Instrument

Item Number	Alpha Coefficient
1	0.829
2	0.829
3	0.825
4	0.833
5	0.830
6	0.821
7	0.837
8	0.825
9	0.830
10	0.831
11	0.830
12	0.829
13	0.830
14	0.826
15	0.821
16	0.834
17	0.818
18	0.819
19	0.820
20	0.817
21	0.828
22	0.828
23	0.832
24	0.820
25	0.835

Appendix K

Supplemental Course Survey Questions from The University of Texas at Austin

Measurement and Evaluation Center

The University of Texas at Austin

Course-Instructor Survey Division BWY310 471-3033 or 1-8175

Supplemental Course-Instructor Survey Items

The Course Instructor Survey System is structured so that instructors can obtain information about their teaching in several ways:

1. The five-item Common Form may be used to obtain general information about a course.
2. One of the 34 standard forms may be used to obtain more specific information about a course.
3. A comment section is included on all answer sheets to allow for student comments.

The CIS system allows for the use of the Common Form and a standard form either alone or in combination. The policies with respect to the choice of forms are set by individual departments/colleges/schools. The materials sent to instructors at the beginning of the semester inviting their participation in the survey contain a list of approved standard forms for various units in the University. In some departments in which participation is required, instructors automatically receive a particular form or combination of forms. Please check with your department/college/school if you are uncertain about policies regarding the use of forms.

Instructors sometimes find, however, that an area of particular concern to them is not addressed on the form(s) they are using. There are two ways in which instructors may obtain information tailored to their specific needs in the Course-Instructor Survey System: 1. The Comments section of the answer sheet may be used to have students respond to essay questions, and 2. Multiple choice questions of the instructor's choosing may be added to the survey.

Comment Section

Instructors may wish to present students with some questions to consider when filling out the Comment section. Some suggestions for this kind of question are:

What in the course has helped you most or do you like the best?

What can the instructor do to improve the course?

What can the instructor do to improve his/her teaching?

Students could also be asked to comment on a particular aspect of the course. Possibilities for such questions include the text, an assignment, or a course topic.

Supplemental Multiple Choice Questions

With over thirty questionnaires in use in the Course-Instructor Survey System, there is considerable variation in the content of the questionnaires. An instructor could easily find that an area of special interest to him or her is not included on the questionnaire(s) chosen. Moreover, instructors sometimes find that the questions on the Common Form or standard forms do not provide information that is specific enough to help them improve their course or teaching. Instructors may find that more detailed questions are of particular use when they are introducing a new course, trying something new in an existing course, or when they are interested in improving their teaching skills.

To assist instructors in tailoring the Course-Instructor Survey system to their individual needs, the Measurement and Evaluation Center has assembled a list of suggested supplemental questions from which instructors may choose. The instructor is responsible for supplying the additional items to the student administrator. The questions can be written on the board, presented on an overhead, or passed out with the regular survey.

The supplemental items have been grouped into major categories used in the field of course evaluation to facilitate their use. It is anticipated that instructors will usually be interested in only a few areas listed here. We hope that these questions may also suggest other possibilities to instructors. The use of any supplemental questions is optional for all instructors.

The answer sheet contains 50 spaces for the responses to the standard form items and any supplemental questions. If a standard form is used, the student responses to any supplemental multiple-choice items will follow those to the standard Course-Instructor Survey questionnaire. The number of supplemental questions that may be added will vary according to the number of questions on the Course-Instructor Survey standard form used. The longest standard form contains 40 questions, so that a minimum of 10 supplemental items may be added. When assembling the items for student use, the number of the first additional question should be the number following the last question on the Course-Instructor Survey standard questionnaire.

The items and response options for the Common Form are printed on the answer sheet in a separate section. If the Common Form only is used, all 50 response options will be available for supplemental items, and the first supplemental item should begin with item 1 in the "Optional items" section of the answer sheet.

Since the response boxes on the answer sheet are labeled A B C D E, the multiple-choice questions can have a range of five choices. The suggested response options for the supplemental questions presented here are:

A Strongly agree
B Agree
C Neutral
D Disagree
E Strongly Disagree

Since the content of the supplemental items is known only to the instructor, the printout of results will contain only the number of each item, with no information regarding item content.

If you have any questions about the use of supplemental Course-Instructor Survey items, please call our office at 471-3033 or 471-8175.

Spring 1988

Course-Instructor Survey
Supplemental Multiple-Choice Questions

Organization of subject matter and course

At the beginning of the course, the instructor gave the class a clear indication of how the course would be organized.
The overall organization of the course facilitated learning.
The lectures and other course activities, such as readings for assignments, worked well together in facilitating learning.
Individual class lectures were presented in an organized fashion.
The lectures were easy to outline.
The instructor used class time well.

Analytic/synthetic approach

The instructor emphasized conceptual understanding.
The instructor explained how course topics were related to each other.
The instructor explained how course topics were related to the main areas in the course.
The instructor contrasted the implications of various theories.
The instructor discussed recent developments in the field.
The instructor discussed points of view other than his/her own.
The instructor helped students integrate prior learning with new material.
The instructor related course material to other areas of knowledge.
The instructor required students to synthesize parts of the course.
The instructor raised challenging questions.

Clarity of presentation

The instructor communicated his/her ideas clearly.
The instructor's classroom presentations were easy to follow.
The instructor stated objectives for each class session.
The instructor was precise in answering questions.
The instructor got right to the point when answering a question.

Helps students distinguish what is important from what is not

The instructor identified the important concepts in the course.
The instructor emphasized what is important in the field.
The instructor summarized major points.
The instructor kept the students focused on relevant material.

Interest in students

The instructor seemed to have a genuine interest in students.
The instructor was fair in his/her dealings with students.
The instructor was considerate in his/her dealings with students.
The instructor seemed sensitive to the feelings of the students.
The instructor seemed sensitive to the needs of the students.

Tolerance of other viewpoints

The instructor was open to other viewpoints.
The instructor invited students to share their knowledge and experience.

(OVER)

COURSE-INSTRUCTOR SURVEY DIVISION - SPRING 1988

Responsiveness to students in instruction

The instructor presented the material on an understandable level.
At the beginning of the course, the instructor determined what skills and abilities students brought to the course.
The instructor clarified thinking by identifying reasons for questions.
The instructor seemed to know when students were bored.
The instructor seemed to know when students were confused.
The instructor used questions, problems, or other methods to check student understanding of the lecture while it was in progress.
After checking to see if the material was being understood, the instructor adjusted the rate of presentation accordingly.
The course allowed adequate individualization of experiences for students with different needs.
The course allowed adequate individualization of experiences for students with different abilities.
I felt I could ask a question during the lecture if there was something I didn't understand.

Approachable

The instructor was readily available for consultation with students.
I felt free to go to my instructor for assistance with problems if I needed to [illegible]

Skill in communication of material

The instructor used clear examples.
The instructor used relevant examples.
The instructor clearly interpreted abstract ideas and theories.
The instructor kept the materials relevant to the course .
The instructor varied the way the material was presented according to what was being taught.

Skill in lecturing

The instructor's speech contributed to his/her teaching effectiveness.
The instructor's lecture style contributed to his/her teaching effectiveness.
The instructor had no speech characteristics that distracted from understanding the lecture.
The instructor varied the speed and tone of his/her voice.
The instructor held my attention during class.
The instructor maintained a classroom atmosphere conducive to learning.
The lecture pace was appropriate.
It was easy to take notes on the lectures.

Skill in directing discussion.

The instructor encouraged class discussion.
The instructor was skilled in directing discussion.
The instructor encouraged students to speak during discussions.
The instructor gave useful feedback on student contributions in discussion.
The instructor kept appropriate control over discussions.
The instructor brought out the main ideas in discussions.
The instructor related ideas discussed to ther class activities and materials.
The instructor used discussion to encourage thinking at higher levels of analysis.
The instructor tried to make it possible for anyone who wanted to participate to have a chance to talk during discussions.
I felt comfortable in participating in the class discussions.

Fairness in examinations and grading

The test questions corresponded to what had been taught.
The examinations permitted students to show understanding.
The examinations accurately measured my achievement in this course.
The examinations were graded objectively.
Enough time was allotted to attempt all the items on the examinations.
I am satisfied with the way the performance of students was evaluated in this course.

Provision of helpful feedback

The instructor used test results to see where students needed extra help.
The instructor commented informatively on my written work, either orally or in writing.
The instructor provided criticism in a constructive manner.
The instructor kept students informed of their progress.

Impact on students

The instructor encouraged students to think for themselves.
The instructor was skilled in bringing out the special abilities of students.
The instructor motivated me to do my best possible work in this course.
I was stimulated to work beyond the requirements of the course.
As a result of this course, I feel more confident in working independently.

Dynamism/enthusiasm

The instructor showed a genuine interest in teaching the course.
The instructor was enthusiastic about the subject.
The instructor seemed to enjoy teaching.

Usefulness of teaching aids

The audiovisuals (slides, movies, transparencies, handouts, guest speakers, or demonstrations) contributed to my learning.

Quality of texts

The text was easy to read.
The text clarified material presented in class
The text was interesting.

Quality of assignments

The assignments were related to the course content.
The assignments helped prepare students for the exams.
The assignments were challenging.
The assignments were interesting.

COURSE-INSTRUCTOR SURVEY DIVISION - SPRING 1988

Appendix L

Final Exam with Posttest Embedded

FINAL EXAM CS315

THE ANSWERS TO PART ONE MUST BE

RECORDED ON SCANTRON SHEETS.

PART TWO IS FREE RESPONSE.

TAKE OFF THIS SHEET AND USE IT FOR SCRATCH PAPER.

QUESTION I. 2 pts each

II.	1.	14 points
	2a.	3
	b.	5
	c.	24

CS315 TO BE FILLED IN ON THE SCANTRON SHEET
Fall 1991

I. Fill in the letter of the correct answer on your scantron sheet.

1. An assertion that states what is true before execution of a code segment.
   a. invariant b. postcondition c. precondition
   d. truth value e. abstraction
2. An assertion that states what is true after execution of a code segment.
   a. invariant b. postcondition c. precondition
   d. truth value e. abstraction
3. An assertion that is always true.
   a. invariant b. postcondition c. precondition
   d. truth value e. abstraction

*4. An ADT where the retrieval operation returns the item that has been in the structure the least amount of time.
   a. list b. stack c. FIFO queue
   d. priority queue e. circle

*5. An ADT where the retrieval operation returns the item that has been in the structure the longest time.
   a. list b. stack c. FIFO queue
   d. priority queue e. circle

*6. How many comparisons are required to sort an array of length 5 if a straight selection sort is used?
   a. 5 b. 10 c. 15 d. 20 e. 25

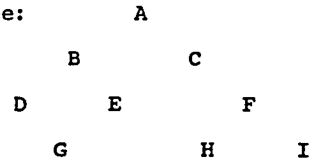
*7. How many comparisons are required to sort an array of length 5 if a straight selections sort is used and the array is already sorted?
   a. 0 b. 1 c. 10 d. 20 e. 30

*8. How many comparisons are required to sort an array of length 5 if the bubble sort with a Boolean flag Switch is used?
   a. 0 b. 1 c. 10 d. 20 e. 30

*9. How many comparisons are required to sort an array of length 5 if the bubble sort with a Boolean flag Switch is used and the array is already sorted?
   a. 0 b. 1 c. 10 d. 20 e. 30

10. Testing based on data coverage.
    a. black box b. regression c. verification
    d. white (clear) box e. validation
11. Testing based on code coverage.
    a. black box b. regression c. verification
    d. white (clear) box e. validation
12. Testing the interfaces among program parts.
    a. black box b. integration c. verification
    d. white (clear) box e. validation

The following questions deal with this tree:



13. Which traversal generates the order: D G B E A C H F I
a. preorder b. postorder c. inorder d. by level e. none

14. Which traversal generates the order: A B G D E C F H I
a. preorder b. postorder c. inorder d. by level e. none

15. Which traversal generates the order: A B C D E F G H I
a. preorder b. postorder c. inorder d. by level e. none

16. If this tree is a binary search tree ordered on a value not shown and the node A is to be deleted, where are the possible replacements for A?
a. node G and node I b. node E and node H
c. node E and node C d. node G and node H
e. node D and node I

*17. If N is the number of data elements to be manipulated by one or more algorithms, then the execution time of each such algorithm can be characterized in terms of N using the Big-O notation. Of the following, which best characterizes an execution time that is significantly different from the others?
a. O(N) b. O(N*N) c. O(N+2)
d. O(N/2) e. O(_¦N)

18. The separation of the logical properties of data from the implementation details.
a. procedural abstraction b. data abstraction
c. data encapsulation d. information hiding
e. primogeniture

19. Six integer numbers are read. Each number is either printed or put on a stack. After the sixth number has been read and processed, the numbers on the stack are popped and printed. If the input values are 1, 2, 3, 4, 5, and 6 (in that order), which of the following lines of output is impossible?
a. 1 2 3 4 5 6 b. 1 3 5 4 6 2 c. 1 3 5 6 4 2
d. none e. all

20. Under the conditions described in question 19, which of the following lines of output is possible?
a. 6 5 4 3 1 2 b. 6 5 4 1 2 3 c. 1 2 4 6 5 3
d. none e. all

21. Six letters are read. Each letter is either printed or put on a FIFO queue. After the sixth letter has been read and processed, the letters on the queue are dequeued and printed. If the input letters are A, B, C, D, E, and F (in that order), which of the following lines of output is impossible?
a. A B C D E F b. A B D C F E c. A C E B D F
d. none e. all

22. Under the conditions described in question 21, which of the following lines of output is possible?
    a. F E D C B A b. A B C F D E c. D E F C B A
    d. none e. all
23. The ability of a program to recover following an error.
    a. verification b. robustness c. abstraction
    d. strength e. primogeniture
24. A special procedure or function that can stand in for a lower-level subprogram when testing.
    a. driver b. recursive procedure c. stub
25. The order in which Pascal stores a two-dimensional arrays.
    a. row major b. column major c. depends on the compiler
26. A queue that keeps on growing.
    a. long queue b. unstable queue c. pool queue
27. Which of the following variables is created at compile time?
    a. global variable b. referenced variable
    c. activation record d. variable on the run-time stack
    e. local variable
28. Which of the following variables is created at run time?
    a. local variable b. referenced variable
    c. activation record d. variable on the run-time stack
    e. all are created at run time
29. The sort whose invariant is:
    Data[1]..Data[I-1] is sorted AND Data[I]..Data[N] is unexamined.
    a. selection sort b. quick sort c. bubble sort
    d. heap sort e. insertion sort
30. The sort whose invariant is:
    Data[1]..Data[I-1] is sorted AND is <= Data[I]..Data[N].
    a. merge sort b. quick sort c. bubble sort w/flag
    d. insertion sort e. is not a proper invariant for a sort
31. The best sort to use when the data is almost sorted.
    a. merge sort b. quick sort c. bubble sort w/flag
    d. heap sort e. insertion sort

*32. In QuickSort, if the splitting value is the first element in the array, then QuickSort will be most efficient with input data in:
    a. sorted order
    b. random order
    c. nearly sorted order
    d. reverse order
    e. law and order

33. The NlogN sort that you should not use if stability is required.
    a. merge sort b. quick sort c. bubble sort w/flag
    d. heap sort e. insertion sort

*34. Merge sort (internal) has which of the following advantages over bubble sort for long lists?
    i. Merge sort requires much less coding to implement than bubble sort
    ii. Merge sort runs faster than bubble sort
    iii. Merge sort requires less memory space than bubble sort
    a. i only b. ii only c. iii only
    d. i and iii e. ii and iii

35. If you want a listing by name within class, you sort first by
    a. class b. name c. it doesn't matter

36. The order of inserting into a list implemented as an ordered linked list.
a. O(1) b. O(logN) c. O(N) d. O(NlogN) e. O(N2)

37. The order of inserting into a list implemented in an ordered array.
a. O(1) b. O(logN) c. O(N) d. O(NlogN) e. O(N2)

38. The order of merging two ordered lists (all implementations are the same).
a. O(1) b. O(logN) c. O(N) d. O(NlogN) e. O(N2)

39. The order of inserting into a priority queue implemented as a heap.
a. O(1) b. O(logN) c. O(N) d. O(NlogN) e. O(N2)

40. The order of creating an empty stack implemented as linked list.
a. O(1) b. O(logN) c. O(N) d. O(NlogN) e. O(N2)

41. The order of creating an empty priority queue implemented as a heap.
a. O(1) b. O(logN) c. O(N) d. O(NlogN) e. O(N2)

42. In a procedure, value parameters do not necessarily protect the contents of the caller's data structures from being affected by execution of the procedure under which of the following conditions?
a. The procedure is recursive
b. The value parameters are integers
c. The value parameters are pointers
d. The value parameters are arrays
e. The procedure is used with a FORWARD declaration

Questions 43 - 48 deal with the following problem.

You have a situation where you need to use two stacks, S1 and S2. You know that together they will never have more than Max elements. You decide to use an array representation with both stacks residing in the same array but with two logical pointers to top, Top1 (for S1) and Top2 (for S2). Notice this will not be a linked list. Use the following definitions and declarations.

```
CONST
  Max = 100;
  MaxPlus1 = 101;
TYPE
  ItemType = (* component on the stack *)
  StackType = RECORD
                 Stack : ARRAY[1..Max] OF ItemType;
                 Top1 : 0..Max;
                 Top2 : 1..MaxPlus1
               END;   (* RECORD *)
```

*43. Choose the statement that correctly implements this procedure.

```
PROCEDURE ClearS2(VAR S2 : StackType);
BEGIN
  (* statement goes here *)
END;
```

a. S2.Top2 := 0
b. S1.Top1 := 0
c. S2.Top2 := MaxPlus1
d. S2.Top2 := Max
e. none is correct

*44. Choose the statement that correctly implements this procedure.

```
FUNCTION Full(S1, S2 : StackType) : BOOLEAN;
(* If S1 and S2 occupy the entire array, Full is TRUE *)
BEGIN
  (* statement goes here *)
END;
```

a. Full := S1.Top1 = Max
b. Full := S1.Top1 = S2.Top2
c. Full := S1.Top1 - S2.Top2 = 1
d. Full := S1.Top1 + S2.Top2 = MaxPlus1
e. none is correct

Questions 45 - 46 deal with the following procedure shell.

```
PROCEDURE PushS1(VAR S1 : StackType; Item : ItemType);
BEGIN
  S1.Top1 := ___________________; (* question 45 *)
  _____________________________ := Item  (* question 46 *)
END;
```

*45. Choose the expression that makes the first statement correct.
a. S1.Top1 + 1
b. S1.Top1 - 1
c. Max - S1.Top1
d. S1.Top1 + MaxPlus1
e. none is correct

*46. Choose the expression that makes the second statement correct.
a. Stack[S1.Top1+1]
b. Stack[MaxPlus1 - S1.Top1]
c. Stack[S1.Top1]
d. S1.Stack[S1.Top1]
e. none is correct

Questions 47 - 48 deal with the following procedure shell.

```
PROCEDURE  PopS2(VAR S2 : StackType; VAR Item : ItemType);
 BEGIN
   Item := ______________________; (* question 47 *)
   S2.Top2 := ___________________  (* question 48 *)
 END;
```

*47. Choose the expression that makes the first statement correct.
a. Stack[S2.Top2]
b. Stack[Max - S2.Top2]
c. S2.Stack[Top2]
d. S2.Stack[S2.Top2]
e. none is correct
*48. Choose the expression that makes the second statement correct.
a. S2.Top2 + 1
b. S2.Top2 - 1
c. MaxPlus1 - Top2
d. Max - S2.Top2
e. none is correct
49. An array contains the following values: P R C Q W B D A T. The values are inserted into a priority queue implemented as a heap. Which is the resulting heap array?
a. W T D R Q B C A P
b. R W D T P B C A Q
c. W T D P R C B A Q
d. W T R Q C B D A P
e. none

Questions 50 - 53 deal with the binary search tree built from entering the following letters P R C Q W B D A T S U (in that order).
50. Which is the successor of P?
a. Q b. W c. T d. P e. S
51. Which is the predecessor of R?
a. R b. D c. A d. Q e. unknown
52. What is the height of the tree?
a. 2 b. 3 c. 4 d. 5 e. unknown
53. If W were deleted, what would be the right child of R?
a. T b. S c. U d. Q e. unknown
*54. Consider the following four tasks:
1. To perform a linear search of a list of N names
2. To perform a binary search of a sorted list of N names
3. To perform a selection sort into alphabetical order of a list of N names that are initially in random order
4. To perform a merge sort into alphabetical order of a list of N names that are initially in random order

For large N, which of the following lists these tasks in order (from least to greatest) or their worst-case running times?
a. 2, 4, 3, 1
b. 4, 3, 1, 2
c. 2, 4, 1, 3
d. 2, 1, 4, 3
e. 2, 3, 1, 4

Questions 55 - 57 use the following declarations.

```
TYPE
  NodePtr = ^Node;
  Node = RECORD
           Info : INTEGER;
           Next : NodePtr
         END;
```

55. Suppose that the only Pascal implementation available does not support pointers or dynamic storage allocation using NEW. You are given a Pascal program that uses pointers and NEW for creating and manipulating linked structures. Your job is to modify the program so that it can be run using the available resources. What should you do?
   a. simulate pointers by using trees and recursion
   b. simulate pointers by using a combination of value parameters and VAR parameters
   c. modify the Pascal implementation so that it does support pointers and NEW
   d. simulate pointers by using indices into a large array and change the program accordingly
   e. identify the queues and stacks in the program, and reimplement each using an array

56. List is an external pointer to a linked list implemented without a header node. If the list is empty, which of the following conditions is both meaningful and true?
   a. List = NIL
   b. List^.Next = NIL
   c. List^.Next = List
   d. List = 0
   e. Next = NIL

57. The separation of the representation of data from the applications that use the data at a logical level.
   a. procedural abstraction b. data abstraction
   c. data encapsulation d. information hiding
   e. primogeniture

*58.

```
function z(k, n : integer) : integer;
begin
  if n = k
     then  z := k
  else
    if n > k
       then  z := z(k, n-k)
       else  z := z(k-n, n)
end;
```

Based on the function defined above, what is the value of z(6,8)?
   a. 1 b. 2 c. 3
   d. 4 e. 8

*59. It would be most appropriate to use a recursive function or procedure to solve a problem that

a. can be reduced to two or more simpler or smaller cases of the same problem
b. involves a substantial number of conditionals and nested loops
c. requires a lot of memory
d. involves storing data in a two-dimensional array
e. involves evaluation of the factorial function

*60.
```
program Main;
 var z:integer;
 function F(x:integer);
 begin
   if (x = 1) or (x = 3)
      then  F := x
      else  F := x * F(x - 1)
 end;
begin  (* main *)
z := F(F(2) + F(5))
end.
```

If MaxInt were large enough to allow the program above to be executed, then at the end of the program, the value of z would be

a. 62
b. 5! + 2!
c. (5! + 2!)!
d. (7!)!
e. (62!)/(2!)

*61. Of the following data representations for storing integers with an arbitrary number of digits, which would allow two integers having that representation to be added and the result to be stored using the same representation in the most time-efficient manner?

a. A binary search tree with the nodes containing the digits and their positions in the decimal representation of the integer and with the nodes ordered by the digits
b. A stack of digits, with the leftmost (most significant) digit on top of the stack
c. A stack of digits, with the rightmost (least significant) digit on the top of the stack
d. A linked list of digits, proceeding from the leftmost (most significant) digit to the rightmost (least significant) digit
e. A linked list of digits, proceeding from the rightmost (least significant) digit to the leftmost (most significant) digit

Questions 62 - 66 use the following array of integers.
20 9 10 3 6 21 8

62. The state of the array after Data[1]..Data[3] is sorted:
9 10 20 3 6 21 8.
Which sort is used?
a. straight selection b. bubble c. heap
d. insertion e. none of these

63. The state of the array after Data[1]..Data[2] is sorted:
3 6 10 20 9 21 8
Which sort is used?
a. straight selection b. bubble c. heap
d. insertion e. none of these

64. The state of the array after Data[1]..Data[2] is sorted:
3 6 20 9 10 8 21
Which sort is used?
a. straight selection b. bubble c. heap
d. insertion e. none of these

65. Quick sort is being used with Data[1] as the split value. What is the state of the array when the first recursive call is made?
a. 6 9 10 3 20 8 21 b. 8 9 10 3 6 20 21
c. 20 9 10 3 6 21 8 d. unknown

66. What is the state of the run-time stack when the first recursive call is made? The base address of the array is 100.

	Data	First	Last	Return	
a.	100	1	7	R1	
	100	1	6	R2	<-- Top of Stack
b.	100	100	106	R1	
	100	100	104	R2	<-- Top of Stack
c.	100	1	5	R2	
	100	1	7	R1	<-- Top of Stack
d.	100	1	7	R1	
	100	1	5	R2	<-- Top of Stack

e. no recursive call is made

Questions 67 - 68 deal with the following problem. The height of a binary tree is the number of nodes in the longest path from the root to a leaf of the tree. The height of an empty tree is 0; the height of a single-node tree is 1.

```
1   Function Height(Tree : TreeType) : INTEGER;
2   Begin
3     If ________________       (* question 67 *)
4       then
5         Height := 0
6       else
7         ______________________     (* question 68 *)
8   end;
```

*67. Fill the blank in line 3.
a. Tree <> NIL b. Tree = NIL
c. Tree^ = NIL d. Tree^.Next = NIL
e. none of the above

*68. Fill the blank in line 7.
a. Height(Tree^.Right) + Height(Tree^.Left)
b. Height(Tree^.Right) + Height(Tree^.Left) + 1
c. Height := Height(Tree^.Right) + Height(Tree^.Left) +1
d. Height := Max(Height(Tree^.Right),Height(Tree^.Left)) +1
e. none of the above

69. The language feature that is necessary for recursion.
a. dynamic variables b. run-time stack c. procedures
d. NEW/DISPOSE e. robustness

Questions 70-74 use the following declarations. You may assume that all scalar data types take one memory location.

```
CONST
  Limit = 100;

TYPE
  ColorType = (blue, white, silver, red, black);
  NameString = PACKED ARRAY [1..15] OF CHAR;
  CarRecord = RECORD
                 Model : NameString;
                 Color : ColorType;
                 Cost : REAL;
                 Doors: (two, four);
                 IDNum: INTEGER;
                 Sold : BOOLEAN
              END;
  Inventory = ARRAY [1..Limit] OF CarRecord;

VAR
  Car : CarRecord;
  InStock : Inventory;
  Index : INTEGER;
```

70. Which of the following expressions accesses the hundreds position of the IDNum of Car?
    a. Car.IDNum[3]
    b. (Car.IDNum MOD 100)
    c. (Car.IDNum DIV 100)
    d. ((Car.IDNum MOD 100) DIV 10)
    e. ((Car.IDNum DIV 100) MOD 10)

71. If Car begins at location 1, what is the location of Car.Model[3]?
    a. 0 b. 1 c. 2 d. 3 e. unknown

72. If Car begins at location 1, what is the location of Car.IDNum?
    a. 3 b. 18 c. 19 d. 20 e. unknown

73. If Car begins at location 1, what is the location of InStock[2].Model[2]?
    a. 24 b. 41 c. 42 d. 27 e. unknown

74. If Car begins at location 1, what is the location of Index?
    a. 2000 b. 2020 c. 2021 d. 2022 d. unknown

Question 75-77 use the following table.

N	column a	column b	column c
32	5	1024	160
64	6	4096	384
128	7	16384	896
256	8	65536	2048
512	9	262144	4608
1024	10	1048576	10240
2048	11	4194304	22528
4096	12	16777216	49152

You have run different algorithms to accomplish the same task and these were the values that were returned as an amount of work for the appropriate value of N (first column).

*75. Which of the following would most closely describe the complexity of the algorithm returning the values in column a?
a. O(1) b. O(logN) c. O(N) d. O(NlogN) e. O(N2)

*76. Which of the following would most closely describe the complexity of the algorithm returning the values in column b?
a. O(1) b. O(logN) c. O(N) d. O(NlogN) e. O(N2)

*77. Which of the following would most closely describe the complexity of the algorithm returning the values in column c?
a. O(1) b. O(logN) c. O(N) d. O(NlogN) e. O(N2)

Name: ______________________________

TA: Sandy Bhala Vince Todd

II. Free response questions

1.

```
   TYPE
     TreeType = ^NodeType;
     NameType = PACKED ARRAY[1..10] OF CHAR;
     NodeType = RECORD
                  Name : NameType;
                  Right, Left : TreeType
                END;   (* RECORD *)
```

Write a recursive function Copy that returns a copy of its input.

```
FUNCTION Copy(Tree : TreeType) : TreeType;
 (* Copy returns a copy of Tree *)
```

2. Consider the problem of designing a data structure to store a list of integers. The structure needs to support only the following operations:

```
MakeEmpty(List)          (* initializes List to empty *)
Insert(List,Value)       (* inserts Value into List *)
Number(List,Value)       (* returns the number of times Value appears
                         in List *)
```

Assumptions:
- The order of values within the list is not important.
- All list values are between 1 and 100.
- No list contains more than 500 elements.

The following schemes are under consideration.

Scheme 1: The list is represented using an array of 100 integers in which the kth element of the array stores the number of times k appears in the list.

Scheme 2: The list is represented using a record containing two fields: an array [1..500] of integers and an integer field containing the number of values currently stored in the list. The kth element of the array stores the kth value inserted into the list.

(a) Choose either Scheme 1 or Scheme 2 and fill in the type declarations for it. Be sure you label clearly which scheme you are using.

```
TYPE
  ListType =                          (* scheme _______ *)
```

(b) Complete the code for Function Number using your scheme.

```
 FUNCTION  Number(List : ListType; Val : INTEGER) : INTEGER;
 (* Number is the number of time Val appears in List *)
```

(c) Fill in the following chart showing the big-O order of the operations for the two schemes.

```
                    Scheme1         Scheme2
               IMMMMMMMMMMMMMMMKMMMMMMMMMMMMMMMM;
MakeEmpty      :               :                :
               GDDDDDDDDDDDDDDDWDDDDDDDDDDDDDDDD6
Insert         :               :                :
               GDDDDDDDDDDDDDDDWDDDDDDDDDDDDDDDD6
Number         :               :                :
               HMMMMMMMMMMMMMMMJMMMMMMMMMMMMMMMM<
```

Appendix M

Pilot Study Pretest

Name: ______________________   SSN: ____________________

Choose the best answer for each of the following questions.

1. Consider the following four tasks:

    1. To perform a linear search of a list of n names
    2. To perform a binary search of a sorted list of n names
    3. To perform a selection sort into alphabetical order of a list of n names that are initially in random order
    4. To perform a merge sort into alphabetical order of a list of n names that are initially in random order

For large n, which of the following lists these tasks in order (from least to greatest) of their worst-case running times?

(a) 2, 4, 3, 1
(b) 4, 3, 1, 2
(c) 2, 4, 1, 3
(d) 2, 1, 4, 3
(e) 2, 3, 1, 4

2. If n is the number of data elements to be manipulated by one or more algorithms, then the execution time of each such algorithm can be characterized in terms of n using the Big-O notation. Of the following, which best characterizes an execution time that is significantly different from the others?

(a) O(N)
(b) O(2N)
(c) O(N + 2)
(d) O(N/2)
(e) O(N)

3. Suppose that the only Pascal implementation available does not support pointers or dynamic storage allocation using NEW. A Pascal program is given that uses pointers and NEW for creating and manipulating instances of the following type.

```
TYPE  Ptr = ^Node;
      Node = record
                info:integer;
                next:Ptr
              end;
```

To modify this program so that it can be run using the available resources, one should

(a) simulate pointers by using trees and recursion
(b) simulate pointers by using a combination of value parameters and var parameters
(c) modify the Pascal implementation so that it does support pointers and NEW
(d) simulate pointers by using indices into a large array, and change the program accordingly
(e) identify the queues and stacks in the program, and reimplement each by using an array

4. Suppose List is a variable of type Ptr that points to the first node of a linear linked list of elements of type

```
Ptr = ^Node;
Node = record
         info:integer;
         next:Ptr
       end;
```

Assume that the list is implemented without a header node. If the list is empty, which of the following conditions is both meaningful and true?

(a) List = NIL
(b) List^.next = NIL
(c) List^.next = List
(d) List = 0
(e) next = NIL

5. In a procedure, value parameters do not necessarily protect the contents of the caller's data structures from being affected by execution of the procedure under which of the following conditions?

(a) The procedure is recursive
(b) The value parameters are integers
(c) The value parameters are pointers
(d) The value parameters are arrays
(e) The procedure is used with a forward declaration

6. Of the following data representations for storing integers with an arbitrary number of digits, which would allow two integers having that representation to be added and the result to be stored using the same representation in the most time-efficient manner?

(a) A binary search tree with the nodes containing the digits and their positions in the decimal representation of the integer and with the nodes ordered by the digits
(b) A stack of digits, with the leftmost (most significant) digit on top of the stack
(c) A stack of digits, with the rightmost (least significant) digit on top of the stack
(d) A linked list of digits, proceeding from the leftmost (most significant) digit to the rightmost (least significant) digit
(e) A linked list of digits, proceeding from the rightmost (least significant) digit to the leftmost (most significant) digit

7. Consider the procedure below. It assumes that the following declarations have been made.

```
TYPE  Ptr = ^Node;
      Node = record
                 info:integer;
                 next:Ptr
              end;
```

First points to the first node, if any, of a linked list.

```
Procedure Unknown(var First:Ptr);
var p, q, r:Ptr;
begin
  p := NIL;
  q := First;
  While q <> NIL do
    begin
      r := q^.next;
      q^.next := p;
      p := q;
      q := r
    end;
  First := p
end;
```

What does the procedure do?

(a) Deletes the first node from the list.
(b) Deletes all but the last node from the list.
(c) Deletes the last node from the list.
(d) Leaves the list unchanged.
(e) Reverses the order of the nodes in the list.

8.

```
Function z(k,n:integer):integer;
 begin
   if n = k then
      z := k
   else
      if n > k then
         z := z(k,n-k)
      else
         z := z(k-n,n)
 end;
```

Based on the function defined above, what is the value of z(6,8)?

(a) 1	(d) 4
(b) 2	(e) 8
(c) 3	

9. It would be most appropriate to use a recursive function or procedure to solve a problem that

(a) can be reduced to two, or more, simpler or smaller cases of the same problem
(b) involves a substantial number of conditionals and nested loops
(c) requires a lot of memory
(d) involves storing data in a two-dimensional array
(e) involves evaluation of the factorial function

10.
```
Program Main;
var z:integer;
Function F(x:integer):integer;
begin
  if (x = 1) or (x = 3) then
     F := x
  else
     F := x * F(x - 1)
end;
begin  (* main *)
  z := F(F(2) + F(5))
end.
```

If MAXINT were large enough to allow the program above to be executed, then at the end of the program, the value of z would be

(a) 62
(b) 5! + 2!
(c) (5! + 2!)!
(d) (7!)!
(e) (62!)/(2!)

11. What value is returned by by the following function given the call WhatIsIt(4,4)?

```
Function WhatIsIt(x,n:integer):integer;
begin
  if n = 1 then
     WhatIsIt := x
  else
     WhatIsIt := x * WhatIsIt(x,n-1)
end;
```

(a) 8
(b) 16
(c) 24
(d) 64
(e) 256

12. Merge sort (internal) has which of the following advantages over bubble sort for long lists?

- i. Merge sort requires much less coding to implement than does bubble sort
- ii. Merge sort runs faster than bubble sort
- iii. Merge sort requires less storage space than bubble sort

(a) i only	(d) i and iii
(b) ii only	(e) ii and iii
(c) iii only	

13. How many comparisons are required to sort an array of length 5 if a straight selection sort is used assuming randomly ordered values?

(a) 5	(d) 20
(b) 10	(e) 25
(c) 15	

14. How many comparisons are required to sort an array of length 5 if a straight selection sort is used and the array is already sorted?

(a) 0	(d) 20
(b) 1	(e) 30
(c) 10	

15. In QuickSort, if the splitting value is the first element in the array, then QuickSort will be most efficient with input data in:

(a) sorted order	(d) reverse order
(b) random order	(e) law and order
(c) nearly sorted order	

Appendix N

Reliability Data for Pilot Study Pretest

Coefficients of Correlation
for Pilot Study Pretest

Item Number	Coef. of Corr. Measure
1	0.42
2	0.51
3	0.16
4	0.33
5	0.32
6	0.58
7	0.57
8	0.43
9	0.61
10	0.56
11	0.51
12	0.19
13	0.53
14	0.61
15	0.53

Overall instrument reliability: 0.73

Bibliography

Austin, H.S. (1987). Predictors of pascal programming achievement for community college students. The Papers of the Eighteenth SIGCSE Technical Symposium on Computer Science Education, 19(1), 161-164.

Ausubel, D.P. (1961). In defense of verbal learning. Educational Theory, 11, 15-25.

Ausubel, D.P., Novak, J.D. & Hanesian, H. (1978). Educational psychology: A cognitive view (2nd ed.). New York: Holt, Reinhart & Winston.

Blosser, P.E. (1981). A critical review of the role of the laboratory in science teaching. (SE 034 834) Columbus, Ohio: ERIC Clearinghouse for Science Mathematics, and Environmental Education.

Blosser, P.E. (1983). What research says: The role of the laboratory in science teaching. School Science and Mathematics Journal, 83(2), 165-169.

Boghai, D.M. (1978). A comparison of the effects of laboratory and discussion sequences on learning college chemistry. Dissertation Abstracts, 39(10), 6045A.

Borg, W.R. & Gall, M.D. (1989). Educational research: An introduction (5th edition). New York & London: Longman.

Bradley, R.H. (1968). Is the science laboratory necessary for general science education courses? Science Education, 52(1), 58-66.

Butcher, D.F. & Muth, W.A. (1985). Predicting performance in an introductory computer science course. Communications of the ACM, 28(3), 263-268.

Campbell, P.F. & McCabe, G.P. (1984). Predicting the success of freshman in a computer science major. Communications of the ACM, 27(11), 1108-1113.

Chavey, D. (1991). A structured laboratory component for the introductory programming course. The Papers of the Twenty-Second SIGCSE Technical Symposium on Computer Science Education, 23(1), 287-295.

Cunningham, H.A. (1946). Lecture demonstration versus individual laboratory method in science teaching--a summary. Science Education, 30(2), 70-80.

Denning, P.J. (chair), Comer, D.E., Gries, D., Mulder, M.C., Tucker, A., Turner, A.J., & Young, P.R. (1989). Computing as a discipline. Communications of the ACM, 32(1), 9-23.

Dey, S. & Mand, L.R. (1986). Effects of mathematics preparation and prior language exposure on perceived performance in introductory computer science courses. The Papers of the Seventeenth SIGCSE Technical Symposium on Computer Science Education, 18(1), 144-148.

Ewbank, W.A. (1971). The mathematics laboratory: What? why? when? how? The Arithmetic Teacher, 18(8), 559-564.

Gathers, E. (1986). Screening freshman computer science majors. ACM SIGCSE Bulletin, 18(3), 44-48.

Grozier, J.E., Jr. (1969). The role of the laboratory in developing positive attitudes toward science in a college general education science course for nonscientists. Dissertation Abstracts, 30(6), 2394A.

Gunsch, L.M. (1972). A comparison of students' achievement and attitude changes resulting from a laboratory and non-laboratory approach to general education physical science courses. Dissertation Abstracts, 33(2), 6290.

Hays, W.L. (1988). Statistics (4th ed.). New York: Holt, Rinehart and Winston.

Hofstein, A., Gluzman, R., Ben-Zvi, R. & Samuel, D. (1980). A comparative study of chemistry students' perception of the learning environment in high schools and vocational schools. Journal of Research in Science Teaching, 17(6), 547-552.

Kirschner, P.A. & Meester, A.M. (1988). The laboratory in higher science education: Problems, premises and objectives. Higher Education, 17, 81-98.

Koffman, E.B., Stemple E. & Wardle, C.E. (1985). Recommended curriculum for CS2, 1984. Communications of the ACM, 28(8), 815-818.

Kurtz, B.L. & Johnson, D. (1985). Using simulation to teach recursion and binary tree traversals. The Papers of the Sixteenth SIGCSE Technical Symposium on Computer Science Education, 17(1), 49-54.

LaGaccia, S.S. (1991). Methodology choices in a cohort of education dissertations. Advances in educational research: Substantive findings, methodological developments, The annual series of the Southwest Educational Research Association, Bruce Thompson (Ed.), 1. (pp. 149-158) JAI Press Inc: Greenwich, Connecticut.

Lawrenz, F. & Munch, T. (1984). The effect of grouping of laboratory students on selected educational outcomes. Journal of Research in Science Teaching, 21(7), 699-708.

Martinez, M.E. & Mead, N.A. (1988). The nations report card: Computer competence the first national assessment. (report no: 17-CC-01). Princeton, New Jersey: Educational Testing Service.

Maxim, B.R. & Elenbogen, B.S. (1987). Teaching programming algorithms aided by computer graphics.

McCracken, D.D. (1989). Three "lab assignments" for an algorithms course. ACM SIGCSE Bulletin, 21(2), 61-64.

Mincy, J.W., Tharp, A.L., & Tai, K.C. (1983). Visualizing algorithms and processes with the aid of a computer. The Papers of the Fourteenth SIGCSE Technical Symposium on Computer Science Education, 15(1), 106-111.

Naps, T.L. (1990). Algorithm visualization in computer science laboratories. The Papers of the Twenty-First SIGCSE Technical Symposium on Computer Science Education, 22(1), 105-110.

National Science Teachers Association. (1991, October/November). Position statement on laboratory science. NSTA Reports! Washington, DC: author.

Odubunmi, O. & Balogun, T.A. (1991). The effect of laboratory and lecture teaching methods on cognitive achievement in integrated science. Journal of Research in Science Teaching, 28(3), 213-224.

Olson, L.J. (1983). A lab approach for introductory programming courses. The Papers of the Fourteenth

SIGCSE Technical Symposium on Computer Science Education, 15(1), 165-167.

Penny, J.P. & Ashton, P.J. (1990). Laboratory-style teaching of computer science. The Papers of the Twenty-First SIGCSE Technical Symposium on Computer Science Education, 22(1), 192-195.

Petersen, C.G. & Miller, N.E. (1985). Open lab vs closed lab: Computer programming I at mississippi state university. The Papers of the Sixteenth SIGCSE Technical Symposium on Computer Science Education, 17(1), 78-81.

Ramberg P. & Van Caster, S. (1986). A new look at an old problem: Keys to success for computer science students. ACM SIGCSE Bulletin, 18(3), 36-37.

Ramlet, J.S. & Folk, M. (1985). PS: A procedure simulator for dynamic program visualization. The Papers of the Sixteenth SIGCSE Technical Symposium on Computer Science Education, 17(1), 36-40.

Roberge', J. & Suriano, C. (1991). Embedding laboratories within the computer science curriculum. The Papers of the Twenty-Second SIGCSE Technical Symposium on Computer Science Education, 23(1), 6-10.

Saunders, W.L. & Dickinson, D.H. (1979). A comparison of community college students' achievement and attitude changes in a lecture-only and lecture-laboratory approach to general education biological science courses. Journal of Research in Science Teaching, 16(5), 459-464.

Schallert, D.L. (1982). The significance of knowledge: A synthesis of research related to schema theory. In W. Otto & S. White (Eds.), Reading Expository Material. (pp. 13-48) New York: Academic Press.

Schriesheim, C.A., Eisenbach, R.J. & Hill, K.D. (1991). The effect of negation and polar opposite item reversals on questionnaire reliability and validity: An experimental investigation. Educational and Psychological Measruement, 51, 67-78.

Sein, M.K. & Bostrom, R.P. (1989). Individual differences and conceptual models in training novice users. Human-Computer Interaction, 4, 197-229.

Taylor, H.G. & Mounfield, L.C. (1989). The effect of high school computer science, gender, and work on success in college computer science. The Papers of the Twentieth SIGCSE Technical Symposium on Computer Science Education, 21(1), 195-198.

Toothacker, W.S. (1983). A critical look at introductory laboratory instruction. American Journal of Physics, 51, 516-520.

Tucker, A.B. (Ed.). (1990). Computing Curricula 1990: Report of the Joint Curriculum Task Force. Review Draft, September, 1990.

Vance, J.H. & Kieren, T.E. (1971). Laboratory settings in mathematics: What does research say to the teacher? The Arithmetic Teacher, 18(8), 585-589.

Weaver, A.C. (1981). Design of a microcomputer laboratory for teaching computer science. The Papers of the Twelfth SIGCSE Technical Symposium on Computer Science Education, 13(1), 132-137.

Webster's college dictionary. (1991). New York: Random House.

Weiner, D.J. (1989). Teaching of assembly language as a laboratory science. ACM SIGCSE Bulletin, 21(4), 60-64.

Wenner, P. (1990). The laboratory component of a computer organization course. The Papers of the Twenth-First SIGCSE Technical Symposium on Computer Science Education, 22(1), 197-200.

Werth, L.H. (1986). Predicting student performance in a beginning computer science class. The Papers of the Seventeenth SIGCSE Technical Symposium on Computer Science Education, 18(1), 138-143.

White, G.M. & Rosenthal, T.L. (1974). Demonstration and lecture in information transmission: A field experiment. The Journal of Experimental Education, 43(2), 90-96.

Wileman, S., Konvalina, J. & Stephens, L.J. (1981). Factors influencing success in beginning computer science courses. The Journal of Educational Research, 74(4), 223-226.

Wilson, C.W. (1991, March). Design, implementation, and evaluation of a cs1 laboratory course. Poster presented at the meeting of ACM SIGCSE, San Antonio, TX.

Wilson, J.D. & Trenary, R.G. (1981). An investigation of computer literacy as a function of attitude. ACM SIGCSE Bulletin, 13(3), 5-12.

Yager, R.E., Engen, H.B. & Snider, B.C.F. (1969). Effects of the laboratory and demonstration methods upon the outcomes of instruction in secondary biology. Journal of Research in Science Teaching, 6, 76-86.

VITA

Debra Lynn Burton was born in Texas City, Texas, on August 24, 1957, the daughter of Robert Charlie Burton and Janis Marie Dugat Burton. She graduated from Clear Creek High School, League City, Texas in 1975. In 1978 she enrolled at Sam Houston State University, Huntsville, Texas from where she graduated in 1980 with a Bachelor of Science degree in Computer Science. While a student at Sam Houston State University she worked twenty hours a week for the computation center, assisting the computer operator, helping students with the operation of the hardware (the computer system was switched from a batch system with key punches and card readers to a multiuser system with terminals while she was employed), and assisting students with programming assignments. After graduation she worked one semester for a C.P.A. firm and then enrolled in graduate school at Southwest Texas State University where she was employed three-quarter time as an assistant instructor in the Department of Mathematics and Computer Science for the two years she was enrolled. In May of 1983 she was awarded a Master of Science degree in Computer Science. After graduation she was employed as an instructor for the Department of Computer Information Systems at Southwest Texas State University. Five years later in 1988, she resigned to pursue her doctorate at

Clemson University, Clemson, South Carolina. After two years at Clemson she transferred into the Science Education Center at The University of Texas at Austin. She was employed for two semesters as a teaching assistant, five semesters as an assistant instructor, and is currently a lecturer for the Department of Computer Sciences at The University of Texas at Austin.

Permanent address: P.O. Box 952
Wimberley, TX 78676

This dissertation was typed by the author.