SIGCSE Bulletin, Vol. 19, No. 1, Feb., 1987, pp. 84-91.

Advanced Placement - Plus in Computer Science
A Summer Institute at The University of Tulsa

Roger L. Wainwright
Dale A. Schoenefeld

Computer Science Department
The University of Tulsa
Tulsa, OK 74104

ABSTRACT

In this paper we discuss an in-service course designed to give secondary school teachers the background needed to teach an Advanced Placement (AP) course in Computer Science. In order to do this effectively, we argue that additional computer literacy and computer system concepts must be developed as well (PLUS). We present the (AP) course outline and objectives as well as the outline and objectives for some additional computing skills.

I. INTRODUCTION

In recent years there has been an enormous increase in computer usage in our educational system at all levels. Many colleges and universities are now requiring at least one computer science course for all students regardless of their major. Secondary schools are increasing their efforts to aquire personal computers and teach some sort of beginning computer course. For a long time the Advanced Placement (AP) Program under the College Board has offered secondary school students an opportunity to study various college-level courses while in high school. Depending on results of the Advanced Placement examination in the particular discipline, a student could receive college credit or advanced standing at a participating college or university. In keeping with the growth in computer science the College Board began offering a Computer Science AP examination in May, 1984. Also, at that time the first edition of the Teacher's Guide was prepared by the AP Computer Science Committee [24].

There has been considerable interest for some time in the Tulsa metropolitan area for developing and teaching the AP computer science course. Many of the teachers who would be responsible for teaching such courses have little or no formal background in computer science. In an attempt to assist teachers in becoming proficient in computer concepts, computer software tools and in the topics covered in the AP computer science course, The University of Tulsa conducted a Teacher Institute in Computer Science from June 9 to July 3, 1986 taught by the authors. The Institute consisted of a four week full-time program emphasizing four major areas: (1) Pascal programming language, programming skills, methodology and data structures (AP), (2) General Computer Literacy, (3) Computer Systems Concepts and (4) Material development and techniques for presenting and teaching computer concepts. Each participant was given an extensive collection of notes and references for future use. In preparing for the Institute several references were consulted in order to benefit from the successes and suggestions of others. Two excellent references that helped in implementing the AP PLUS course are found in [18,23].

II. COURSE PHILOSOPHY AND IMPLEMENTATION

The course content was divided into two parts; the AP part and the PLUS part. The AP part consisted of the AP course as proposed by the Computer Science Development Committee. The PLUS part consisted of numerous topics in computer literacy and computer system concepts. The PLUS part of this course is a necessary and vital component and is what makes this course different from many other AP courses. Although the AP part was the primary emphasis (composing approximately 70% of the time), the PLUS component added the needed depth and understanding of fundamental computer concepts and software tools that many secondary school students will want to learn. It also will enable the course participants to be valuable resource people in their school environments. On the first day of class the participants were queried to determine their awareness of computer literacy and computer system concepts. On that basis the content of the PLUS part of the Institute was developed.

Even though the presentation was intense, four weeks seems to be the right amount of time to cover all of the AP and PLUS material. The daily schedule for Monday thru Thursday consisted of an AP lecture from 8:30 to 10:45am with a short break in the middle followed by a PLUS lecture from 11:00 to noon. On Friday an examination was given from 8:30 until approximately 10:15 followed by an AP lecture from 10:30 until noon. Daily lab time was assigned from 1:00 to 3:00pm on each day, although most people stayed until 3:30 or even later. This arrangement of time seemed to work quite well. The course was team taught. One instructor taught all of the AP material while the other was responsible for all of the PLUS material. A graduate assistant was assigned to help supervise the lab work. The assistant was a former high school teacher on leave of absence to finish his graduate degree in computer science and had previously taught the AP course in high school. This schedule allowed the participants daily exposure to lectures on two different topics. This made the long lecture time in the morning go much faster.

Computing facilities were provided by the Computer Resource Center at The University of Tulsa. The laboratory used contained 30 IBM PC's and 20 Apple MacIntosh computers. During the four weeks of the Institute the laboratory was open from 10:00am to 8:00pm on weekdays to all university students including the Institute participants. Approximately half of the laboratory facilities were reserved during the assigned lab hours from 1:00 to 3:00pm daily. Thus each student participating in the Institute had access to his own PC. The PC's were all located in the same room with white board and TV monitor facilities in case some additional explanations were needed. Students could have been assigned to CRT terminals and given access to the university's mainframe computer facilities. However, the facilities currently being used in local and regional high schools are generally personal computers of some kind. Thus the orientation of the Institute was toward PC users.

Lab assignments were given covering both the AP and PLUS topics. Most lab assignments, however, were oriented toward Pascal and programming techniques.

At all times the lab assistant and at least one of the instructors were available for programming assistance. This was more than adequate support. Lab assignments were coordinated with the morning lecture topics. Typically each assignment took from two to three days to complete. However, assignments overlapped, and while students were finishing one assignment they were beginning another. A list of the lab assignments is given in Appendix III and Appendix IV.

III. INSTITUTE PARTICIPANTS AND ADMINISTRATIVE ISSUES

Eighteen students participated in the Computer Science Institute. They represented ten different school districts and thirteen different schools from the Tulsa metropolitan area and surrounding region. Several students commuted a short distance of 3Ø miles or so everyday. One student commuted from a distance of 6Ø miles daily and two other students commuted weekly from a distance of 2ØØ miles. Thus one of the goals that this be a regional Institute was achieved. Student background varied quite a bit. Although all students had some prior exposure to computing, it ranged from FORTRAN IV 15 years prior to one student having taught the computer science AP course twice. Only three students had any prior exposure to Pascal and in each case it was self taught. For the majority of the students all of the material presented in the AP and PLUS parts was completely new. All students had some prior computing experience, generally programming in BASIC. This was not a prerequisite for entrance into the Institute, but it did allow for the presentation of more material during the four weeks. It is recommended, if possible, that some prior exposure to programming be a requirement when putting together an AP-PLUS course, otherwise one would not have time to cover the PLUS part. Each student enrolled in the course for academic credit (six graduate hours). Only one student choose to audit the course and did not take the tests. The University of Tulsa provided a tuition scholarship for each participant and in addition each student was given a $4ØØ stipend. The stipend was paid upon completition of the course and was more than enough to cover the cost of textbooks and ,if needed, one months rent at one of the university apartments. Attendance was perfect, except one student was ill for one day. All of the eighteen students finished the course.

IV. COURSE MATERIALS, CONTENT and GRADING

Three textbooks were adopted for use in this course. Introduction to Pascal and Structured Design by Dale and Orshalick [2Ø] was selected for the Pascal language portion of the AP course. Pascal plus Data Structures, Algorithms and Advanced Programming by Dale and Lilly [19] was selected for the data structures portion of the AP course. Using Microcomputers, Tutorials for dBASE II, WORDSTAR, and 1-2-3 by Brightman and Dimsdale [15] was adopted for part of the PLUS course.

The criteria for selecting the two AP textbooks included (1) providing teachers with a good source of reference material after the course was over, (2) readability by students after having heard a lecture on a topic, (3) consistency in style of presentation, approach and content, (4) availablility of chapter exercises, sample tests and programming exercises. Although there are several good texts that meet these requirements, the two texts by Dale were selected. The Dale and Orshalick text has a section on teaching and debugging at the end of each chapter. There are also chapter exercises and a pre-test at the end of each chapter as well. It has an excellent glossary of computer terms along with answers to the chapter excerises. There is also a small supplement for instuctors containing programming exercises for each chapter. There is an additional instructors guide containing, among other items, the answers to all of the chapter pre-tests. These materials were made available to the students. The Dale and Lilly text is arranged in a similar fashion. Each chapter has a summary, a typical application using the concepts in the chapter, excerises and a pre-test. There are also instructor's guides with solutions to exercises and pre-tests. For a detailed outline of the material presented in the AP part and the chapters coverd in these textbooks, see Appendix I.

The topics covered during the PLUS part included (a) the Microsoft Disk Operating System, (b) the development environment of Turbo Pascal with Wordstar editing, (c) electronic spreadsheets and graphics using Lotus 1-2-3, (d) machine language programming for the Intel 8Ø88, (e) assembler language programming using Microsoft 8Ø88 Macro Assembler, (f) disk organization on an MS DOS diskette, (g) computer graphics using the Turbo Graphix Toolbox, (h) database management systems and relational algebra queries using R:Base 5ØØØ, (i) custom database application systems written with the Turbo Pascal Database Toolbox, (j) the Apple Macintosh user interface illustrated using Jazz, and (k) the Apple Macintosh Quickdraw ROM routines accessed from MacPascal.

Each of the participants was requested to indicate his entering level of awareness for each of the above PLUS topics. Sixty-eight percent (68%) of the responses were "no awareness," twenty-nine percent (29%) of the responses were "vague awareness," and three percent (3%) of the responses were "substantial awareness." The three percent "substantial awareness" responses represented only four responses and they were from different participants on different topics.

Selecting a single textbook for the PLUS part of the course was not possible given the breadth of the objectives. Using Microcomputers, Tutorial for dBase II, Wordstar, and 1-2-3 by Brightman and Dimsdale [15] was a useful self-contained presentation for some of the topics but it was not sufficient for communicating the desired perspective on all topics. As implied by the title, the Brightman and Dimsdale book is tutorial in orientation. Sample data files consistent with the text presentation were available in electronic form. However, licensed system software necessary to manipulate the data files of Brightman and Dimsdale and system software to meet the other objectives of the PLUS component were made available through the Computer Resource Center. The software included the IBM Disk Operating System 3.1; Turbo Pascal, Turbo Graphix Toolbox, and Turbo Database Toolbox from Borland International; Lotus 1-2-3 and Jazz from Lotus Development Corporation; R:Base 5ØØØ from Microrim; and MacPascal from Apple Computer, Incorporated. In addition to the tutorial and reference documentation that accompanies the indicated software products, other book materials that were consulted extensively and overviewed during the PLUS lectures are listed in the bibliography and are included in the course outline of Appendix II.

Each examination consisted of two parts. Approximately 75% covered material from the AP course and 25% from the PLUS course. The first exam was a closed book test since it covered the fundamentals of the Pascal language. In the remaining examinations students were allowed to use their books, notes and handouts. The course grade was determined by the following weights: 7Ø% from the performance on the AP exams, 2Ø% from the PLUS exams and 1Ø% from the lab assignments.

V. CONCLUSIONS

We feel the AP-PLUS course was very successful. We were surprised at the continued good attitude and the level of hard work put forth by all of the participates even though we worked them very hard. A significant reason for this was the ability to attract quality participates from our region by providing full tuition and a stipend. Our decision to include the PLUS part into the course was well received and proved to be a significant addition to the traditional AP course. Appendix V gives a summary of a participant survey over the PLUS part of the course. Most participates had either no or vague background in each of the various topics. In nearly every case most participates appreciated the extra course material as a supplement to the main line AP part of the course.

VI. ACKNOWLEDGEMENTS

The AP-PLUS Computer Science Institute was funded by the Oklahoma State Board of Regents of Higher Education. These funds allowed us to attract quality participants by providing a small stipend and provide the necessary personnel and materials to develop the curriculum. The Institute funding was also supplemented by The University of Tulsa which provided a tuition scholarship (six graduate hours) for each participant.

BIBLIOGRAPHY

[1] ---, Disk Operating System, Version 3.1, software and reference materials, IBM Corporation, 1984.

[2] ---, Inside the Macintosh, Volumes I, II, III, Addison Wesley, 1985.

[3] ---, Lotus 1-2-3, Release 2, software and reference materials, Lotus Development Corporation, 1985.

[4] ---, Lotus Jazz, software and reference materials, Lotus Development Corporation, 1985.

[5] ---, The Lotus Guide to Learning Lotus 1-2-3, Includes Release 2, Lotus Development Corporation, 1985.

[6] ---, The Lotus Guide to Jazz, Lotus Division of Addison Wesley, 1985.

[7] ---, Macintosh Pascal 2.Ø, software and reference materials, Apple Computer, Inc., 1985.

[8] ---, Microsoft Macro Assembler Package, software and reference materials, Microsoft Corporation, 1984.

[9] ---, R:BASE 5000, software and reference materials, Microrim, 1985.

[10] ---, R:BASE 5000 Tutorial, subset system software and tutorial materials, Microrim, 1985.

[11] ---, Turbo Database Toolbox, software and reference materials, Borland International, 1985.

[12] ---, Turbo Graphix Toolbox, software and reference materials, Borland International, 1984.

[13] ---, Turbo Pascal, Version 3.0, software and reference materials, Borland International, 1985.

[14] Arganbright, Mathematical Applications of Electronic Spreadsheets, McGraw Hill, 1985.

[15] Brightman, Richard W. & Dimsdale, Jeffrey M., Using Microcomputers, Tutorials for dBase II, Wordstar, and 1-2-3, Delmar Publishers Inc., 1986.

[16] Chernicoff, Macintosh Revealed, Volume One, Unlocking the Toolbox, Hayden Press, 1985.

[17] Chernicoff, Macintosh Revealed, Volume Two, Programming with the Toolbox, Hayden Press, 1985.

[18] Comer, James R. and Schember, Kurt A. "Advanced Placement in Computer Science: A Summer Workshop", ACM SICCSE Bulletin, Vol. 16, No. 1 (February, 1984), pp 119-123.

[19] Dale, Nell and Lilly, Susan C. PASCAL and Data Structures, Algorithms and Advanced Programming, D. C. Heath and Company, 1985.

[20] Dale, Nell and Orshalick, David. Introduction to PASCAL and Structured Design, D. C. Heath and Company, 1983.

[21] Date, C. J., An Introduction to Database Systems, Fourth Edition, Addison Wesley, 1986.

[22] Foley and Van Dam, Fundamentals of Interactive Computer Graphics, Addison Wesley, 1982.

[23] Frederick, William G. and Mansfield, Maynard J. "Introductory Computer Science Courses for In-Service Elementary and Seconary Teachers", ACM SIGCSE Bulletin, Vol 17, No. 1 (March, 1985) pp 211-214.

[24] Garland, Stephen J., et al. editors. College Board's Advanced Placement Computer Science Development Committee, Teacher's Guide to Advanced Placement Courses in Computer Science, College Entrance Examination Board, 1983.

[25] Hopgood, et al, Introduction to the Graphic Kernal System (GKS), Academic Press, 1983.

[26] Koffman, Elliot, Turbo Pascal, A Problem Solving Approach, Addison Wesley, 1986.

[27] Kronick, Macintosh Pascal Illustrated - The Fear and Loathing Guide, Addison Wesley, 1985.

[28] Lemone, Karen, Assembly Language and Systems Programming for the IBM PC and Compatibles, Little Brown, and Company, 1985.

[29] Ledgard and Singer, Pascal for the Macintosh, Addison Wesley, 1986.

[30] Lu, The Apple Macintosh Book, Microsoft Press, 1984.

[31] Norton, Peter, Programmer's Guide to the IBM PC, Microsoft Press, 1985.

[32] Radford & Haigh, Turbo Pascal for the IBM PC, Prindle, Weber and Schmidt, 1986.

[33] Simpson, Alan, Understanding R:BASE 5000, Sybex, 1985.

[34] Ullman, Jeffrey, Principles of Database Systems, Second Edition, Computer Science Press, 1982.

[35] Van Wolverton, Running MS DOS, Microsoft Press, 1985.

APPENDIX I. AP COURSE OUTLINE

TEXTS: Introduction to PASCAL and Structured Design by Nell Dale and David Orshalick, D. C. Heath and Company, 1983. [20] (D & O)

PASCAL Plus Data Structures, Algorithms and Advanced Programming, Nell Dale and Susan C. Lilly, D. C. Heath and Company, 1985. [19] (D & L)

AP COURSE OUTLINE

DAY	LECTURE TOPIC	READING ASSIGNMENT
Week 1 (Mon)	Overview of programming, computing and languages. Introduction to PASCAL, syntax diagrams, assignment statements, expressions simple I/O, program construction.	Chapters 1,2 (D & O).
Week 1 (Tue)	IF-THEN-ELSE, Top-down design, Methodology, Boolean expressions, Truth Tables, Operator	Chapters 3,4 (D & O).

	precedence, ORD, PRED, SUCC, ASCII internal representation.	
Week 1 (Wed)	Functions, Procedures, Scope rules, Argument and parameters, Local and Global, Passing by VAR and copy value.	Chapters 6,7 (D & O).
Week 1 (Th)	Additional control structures. Looping and counting loops. WHILE, REPEAT, CASE, FOR statements	Chapters 5,8 (D & O).
Week 1 (Fri)	** TEST 1 ** More on input of characters. EOLN, EOF. SETS in PASCAL	Chapter 13 (D & O).
Week 2 (Mon)	REVIEW- Put the pieces together. A problem is presented, pseudo code is written in a top down fashion. Finally Pascal code is written using all features discussed to date	
Week 2 (Tue)	Ordinal types, Data types, Subrange types, Data structures, Arrays, Packed and Unpacked	Chapter 9 (D & O).
Week 2 (Wed)	Records, WITH, Arrays of records, Hierarchical (nested) records. Variant records with and without a TAG field.	Chapter 12 (D & O).
Week 2 (Th)	Pointer variables and linked lists. DISPOSE, NEW.	Chapter 14 (D&O). Chapters 5,6 (D & L).
Week 2 (Fri)	** TEST 2 ** Pointers and linked lists continued... Insertion and deletion from an ordered and unordered list. Double linked and circular linked lists.	Chapter 14 (D&O). Chapters 5,6 (D & L).
Week 3 (Mon)	Functions continued and introduction to Recursion and recursive procedures and functions.	Chapter 8 (D & O) Chapter 7 (D & L)
Week 3 (Tue)	Recursion continued... Factorial, Towers of Hanoi and several other problem to trace. Sample practical problems to solve.	Chapter 8 (D & O) Chapter 7 (D & L)
Week 3 (Wed)	Sorting and Searching Algorithms Using Arrays. BIG O Notation for determining the order of an algorithm	Chapters 11,12 (D & L).
Week 3 (Th)	STACKS and QUEUES as abstract data types with some simple problems	Chapters 3,4 (D & L).
Week 3 (Fri)	** TEST 3 ** STACKS and QUEUES continued...	Chapters 3,4 (D & L).

	Implementing STACKS using either linked list or arrays. Use stacks to convert infix to polish and polish to infix notation.	
Week 4 (Mon)	Introduction to TREES. Binary Search Trees. Definitions of various Tree traversals. Level order traversal using a queue. Inorder and other traversals implemented using a stack and using recursive procedures.	Chapter 8 (D & L)
Week 4 (Tue)	Trees continued... Heaps and Heapsort implemented using arrays not pointers.	Chapters 8,9 (D & L).
Week 4 (Wed)	Misc. topics not tested over including Computer Science Curriculum issues and an introduction to parallel processing and parallel computers.	
Week 4 (Th)	** FINAL EXAM **	
Week 4 (Fri)	(no class due to July 4 holiday)	

APPENDIX II PLUS COURSE OUTLINE

TEXT: Brightman, Richard W. and Dimsdale, Jeffrey M., Using Microcomputers, Tutorials for dBase II, Wordstar, and 1-2-3, Delmar Publishers, Inc., 1986. [15]

DAY	LECTURE TOPIC	READING/REFERENCE
Week 1 (Mon)	The Microsoft Disk Operating System: To use introductory DOS commands, to use DOS filters and pipelining, to use the hierarchial file system to manage hard disks, to create command files.	pp 40-68 [15], [1], [35]
Week 1 (Tue)	The Development Environment of Turbo Pascal with Wordstar Editing: To use the Wordstar subset editor embedded in Turbo Pascal, to edit, compile, link, and execute a Pascal program using Turbo Pascal.	pp 122-181 [15], pp 7-35 [13], pp A9-A15 [26], pp 1-31 [32]
Week 1 (Wed-Th)	Electronic Spreadsheets and Graphics Using Lotus 1-2-3: To understand the general capabilities of electronic spreadsheets, to navigate sufficiently well with Lotus 1-2-3 to build an introductory spreadsheet, to create graphics output using Lotus 1-2-3, to note that numerous problems in many disciplines are ameanable to solutions with an electronic spreadsheet, to write a Lotus macro, to see an amortization schedule, to see an income tax table application of lookup, to graph x*sin(x).	pp 182-266 [15], [3], [5], [14]

Week 1 (Fri)	** TEST 1 **	
Week 2 (Mon)	Machine Language for the Intel 8088: To indicate strategies for internal representations of character, integer, and real data (that is, ASCII, 2's complement, IEEE short real format), to understand the basic architecture of a computer system, to use debug to enter and execute an 8088 machine language program that generates the sum of of two integers.	[1], [8], [28], [31]
Week 2 (Tue)	Assembler Language Using Microsoft Macro Assembler: To enter, assemble, link and execute an 8088 assembler program that generates the sum of two integers.	[1], [8], [28], [31]
Week 2 (Wed)	Disk Organization for an MS DOS Diskette: To see what, where, and how information is stored on an IBM double side, double density, 40 track diskette with discussion on the header tracks, the file allocation table, and the directory; to gain an appreciation for the objective of disk recovery packages like Norton Utilities.	[1], [8], [31]
Week 2 (Th)	Computer Graphics Using the Turbo Graphix Toolbox: To be introduced to the Turbo Graphix Toolbox - a library of routines for the IBM PC with a color graphics card or a Hercules monochrome graphics card, to understand the motivation and implementation of some introductory graphics algorithms including window-viewport mapping and clipping, to become aware of the Graphics Kernal Standard as a strategy for providing functionality and portability.	[12], [22], [25]
Week 2 (Fri)	** TEST 2 **	
Week 3 (Mon, Tue, Wed)	Database Management Systems and Relational Algebra Queries Using R:Base 5000: To To define the structure of a database, to enter data into a database, and to produce reports from a database using R:Base 5000; to convey the expressive power of the relational operators including union, difference, cartesian product, projection, selection, intersection, quotient, and join operating on tables to produce responses to queries; to observe that the R:Base 5000 implementation of the operators is non-standard.	pp 69-122 [15], [9], [10], [21], [33], [34]

Week 3 (Th)	Custom Database Application Systems written using the Turbo Database Toolbox: To illustrate the advantage of incorporating existing software libraries into application development in Pascal or other high level languages, to illustrate the program and algorithm sophistication (e.g. B+ Tree, ISAM file management routines in the Database Toolbox) required to efficiently accomplish objectives that the naive user may take for granted.	[11]
Week 3 (Fri)	** TEST 3 **	
Week 4 (Mon)	The Apple Macintosh User Interface Illustrated with Lotus Jazz: To become oriented to the mouse/icon user interface represented by current generation computer products including the Apple Macintosh, to become aware of the advantages of a software package that integrates various productivity tools, to be able to navigate with Jazz in order to create a document that contains spreadsheet information, graphics, and ordinary text.	[4], [6], [3Ø]
Week 4 (Tue-Wed)	The Apple Macintosh Quickdraw ROM Routines Accessed from MacPascal: To be aware of the pedagogical advantage of MacPascal and its multiwindow environment (to stepwise trace the execution of a Pascal program while viewing graphics output, text output, and variable trace information in different windows of the same screen display) for teaching Pascal, to understand the capabilities of Quickdraw - a collection of graphics routines implemented in a Macintosh ROM and playing an active role in all Macintosh screen activity, to hint at procedures and facilities for doing system development on the Apple Macintosh.	[2], [7], [16], [17], [27], [29]
Week 4 (Th)	** FINAL EXAM **	

APPENDIX III. AP LAB ASSIGNMENTS

LAB AP.1

Objectives: To learn the Turbo Pascal operating envirnonment
To learn Pascal syntax and write simple programs

Two partially written programs were given to the students to complete. Write a program to calculate N to the power of M, where N and M are read in from the terminal. Write a second program that takes a salary as input and determines the amount to be deducted assuming that a tax table is provided along with several other deductions including number of children and married or not married. Write a third program to find the minimum number of rolls of coins (quarters, dimes, nickels, pennies) in a given amount of money. Determine the minimum number of coins in any amount of money left over. (Program outlines were provided for the first two programs).

LAB AP.2

Objectives: To use Packed Arrays
To use loop constructs such as FOR, REPEAT-UNTIL, WHILE
To use EOLN and EOF as boolean operators
To create and read a disk file using TURBO file I/O conventions for text files.

Write a program which reads, character by character, a text file located on disk, and prints each word read in along with its length. A word is defined as any sequence of characters in A..Z. A word is delimited by any character not in A..Z. Write this program using several procedures.

LAB AP.3

Objectives: To use character arrays
To use the CASE statement
To further develop programming skills in Pascal

Write a Pascal program to isolate words from a text file on disk. Record each different word of length 1, 2, 3 or 4 and the frequency of each. Assume no more than 5Ø different such words. Print out your results. Use the procedure you wrote in LAB 2.

LAB AP.4

Objectives: To analyze the running times of an algorithm using the internal clock on the PC.
To make use of a random number generator
To discover how bad bubble sort is for large N.

Write a Pascal program that will generate N random keys in the range of 1..1ØØØ and perform a bubble sort on the N keys. Input the value of N from the terminal. Try N=5ØØ and N=1ØØØ at least. A procedure verifying the keys are indeed sorted should be included. Output the time to sort. Also predict how long it would take to sort one million keys.

LAB AP.5

Objectives: To make use of recursion in Pascal
To discover the difference in execution times required to solve a problem using recursion verses iteration.

Write two procedures to determine the N th FIBONACCI number in a sequence that is seeded with Ø and 1. One routine should be recursive and the other is to be iterative. Input N from the terminal and output the running times for each routine. Use N = 1Ø,15,2Ø,25 and 3Ø.

LAB AP.6

Objectives: To use pointer variables and records
To use recursive procedures
To discover and compare an O(N log N) algorithm verses an

O(N*N) algorithm from LAB 4.

Write a Pascal program to sort N randomly generated keys in the range of 1..20000 by inserting them into a Binary Search Tree and using an inorder (symmetric) traversal to read out the result in sorted order. Input N from the terminal and output the execution time. Use N = 200, 1000 and predict the time to sort one million keys.

APPENDIX IV. PLUS LAB ASSIGNMENTS

LAB PLUS.1

Objectives: To apply electonic spreadsheet software with graphics capabilities in a typical mathematics graphing context.
To understand the general capabilities of electronic spreadsheets and to suggest creative ways for their use.

Use Lotus 1-2-3 to build an electonic spreadsheet that will allow the user to specify a continuous bounded function, including its domain, and to subsequently invoke a macro to draw a "smooth" curve connecting 200 ordered pairs on its graph. Print the graph to an IBM Graphics Printer and to a Hewlett-Packard pen plotter.

LAB PLUS.2

Objectives: To apply database software to a frequently required business activity.
To appreciate the high utility and ease of use of commonly available database software (that is, one can be productive without being Pascal literate.)

Use R:Base 5000 to build a name-address database. In zip code order, print a form letter and a mailing label for each individual in the database. In alphabetical order on last name, print a report listing all entries in the database.

LAB PLUS.3

Objectives: To build a MacPascal program that produces text, graphics, and sound output on the Apple Macintosh.
To gain experience with the popular Apple Macintosh and to appreciate the uniqueness of its bit mapped graphics orientation.

Draw a single octave piano keyboard that can be audibly played, single key at a time, by first selecting a note with the mouse driven cursor and then clicking the mouse button.

APPENDIX V. PARTICIPANT SURVEY

Thirteen of the participants completed a survey administered immediately following the final examination. The survey was intended to measure their entering awareness level on the PLUS topics and to judge their reaction to the overview presentations made during the Institute on the PLUS topics. This appendix is a summary of their responses.

	Prior Awareness			Reaction to Overview Presented	
PLUS TOPIC	None	Vague	Substantial	Appreciated	Could Not Care Less
MS Disk Operating System	8	4	1	12	1
Turbo Pascal Operating Environment	9	3	1	13	Ø
Electronic Spreadsheets	7	6	Ø	12	1
Machine Language Programming	7	6	Ø	6	6
Assembler Language Programming	6	7	Ø	7	6
MS DOS Diskette Organization	7	6	Ø	11	2
Computer Graphics	4	8	1	11	2
Relational Database Management Systems	12	1	Ø	11	2
ISAM File Management Library Routines	12	1	Ø	8	5
Macintosh Architecture/ User Interface	12	1	Ø	1Ø	3
Macintosh Quickdraw Rom Routines/Pascal Prog	11	1	1	11	2