HERBERT D. PECKHAM

BASIC

A Hands-on Method

```
370  REM COMPUTE PRICE FOR GRADE B
380  GOSUB 800
390  PRINT "B",FNR(P)
400  REM COMPUTE PRICE FOR GRADE C
410  GOSUB 800
420  PRINT "C",FNR(P)
430  REM COMPUTE PRICE FOR GRADE D
440  GOSUB 800
450  PRINT "D",ENR(P)
460  STOP
```

15

BASIC

A Hands-on Method

HERBERT D. PECKHAM

Professor of Natural Science
Gavilan College

McGraw-Hill Book Company

New York St. Louis San Francisco Auckland Bogotá Düsseldorf Johannesburg
London Madrid Mexico Montreal New Delhi Panama Paris
São Paulo Singapore Sydney Tokyo Toronto

Library of Congress Cataloging in Publication Data

Peckham, Herbert D
BASIC: a hands-on method.

Includes index.
1. Basic (Computer program language) I. Title.
QA76.73.B3P4 001.6'424 77-24340
ISBN 0-07-049146-1

BASIC: A Hands-on Method

2 3 4 5 6 7 8 9 0 K P K P 7 8 3 2 1 0 9 8

This book was set in Caledonia by Creative Book Services, subsidary of McGregor & Werner, Inc. The editor was Peter D. Nalle; the designer was Elliot Epstein; the production supervisor was Milton J. Heiberg.
Kingsport Press, Inc., was printer and binder.

BASIC

A Hands-on Method

Preface

This book grew out of a sense of frustration with existing BASIC programming texts intended for liberal arts students. Two characteristics of most of the texts on the market are most objectionable. First, almost all quickly begin to use mathematics at a level that excludes the vast majority of the very students we are most interested in, many of whom can rely on introductory algebra (very dimly remembered) but who need to learn how to program in BASIC. The second objection is that generally nothing in the structure of the texts requires students to spend much (if any) time on the computer. Students typically try to study programming like any other subject and do not feel the need to experiment with and execute programs on the computer. It seems axiomatic that much more effective learning will take place if most of the study of BASIC takes place on the computer. This text's main thesis is that more traditional text material should be preceded by a good deal of time experimenting with the language on the computer. The experience to date validates the idea that students work through the material more rapidly and effectively with this initial exposure to BASIC on the computer.

The instructor and student will immediately note that the structure of this book is quite different from that of most texts on the market. Each chapter begins with a statement of the objectives for that chapter. Then students are guided through a set of exercises that let them experiment with the characteristics of BASIC and see the language in action. Once students have acquired a "feel" for BASIC, they can profitably proceed to a more traditional treatment. The mathematics level has intentionally been kept very low. Students with more advanced mathematical skills can learn to use these on the computer on their own. However, if the mathematics level in the text were set too high, the majority of beginning students would become discouraged in the first few chapters. At the level presented, nearly any student should be able to work through the material without getting "hung up" by the mathematics. Students must have access to a time-sharing computer that supports the BASIC language.

The text is organized into nine chapters plus an appendix. Each chapter forms a module of instruction that should require about 2 hours of classroom time and possibly 3 or 4 hours of time outside class. Review tests are provided at the end of each chapter, enabling the student to see if the objectives have been mastered. A key explaining how to use the text material on various computers is given in Appendix A.

The text can be used in several different ways. It has been used very effectively in an open-entry, open-exit, self-paced course. If desired, the material can be presented in a traditional lecture format. Finally, it can be used with a minimum of supervision as a self-study text.

Students at any level from junior high through graduate school should be able to use the material effectively. The goal is to provide programming skills in BASIC as rapidly and effectively as possible. As indicated above, no mathematics past introductory algebra is required, and the algebra used is mainly formula evaluation. More mathematical ability is nice but unnecessary.

FOR THE INSTRUCTOR

Students must have account numbers to gain access to the computer. Certainly, the best method is to issue a different account number to each student. If desired, it is usually possible to issue a single account number to an entire class. The issue should be explored with the computer staff and settled one way or the other.

There are several advantages to having individual account numbers. When students request a listing of programs they have stored, they will see only the programs they have put there. However, if a common account number is used, all programs stored by the class will be in the same location on the computer. This may cause some confusion for the beginner. Most computers have programs that print out terminal time accumulated for each account number. If individual account numbers are used, this record provides an easy way to keep track of the time spent by individuals on the computer.

Allow about 1000 characters of storage in memory for each student. If a group account is used, all this memory allocation should be lumped together. Some of the computer exercises require students to move programs to memory and then retrieve them. This is the purpose of the storage space in memory.

You should check Appendix A to see if your computer is listed. If so, verify that instructions for various system commands are correct. If your computer is not included, you can easily prepare a student handout giving the proper procedures for your facility using Appendix A as a guide.

ACKNOWLEDGMENTS

The author is deeply grateful for several most valuable sources of assistance. Several dozen community college instructors in northern California have used preliminary versions of the material and provided most useful suggestions for improvement. A special vote of thanks must go to a colleague at Gavilan College, Professor John Hansell, who read the manuscript and helped eliminate the "buzz words" that crept in. Yvonne Wingo provided much needed assistance with graphics. The errors that remain are, of course, due to me.

HERBERT D. PECKHAM

Contents

Preface xi

1 INTRODUCTION TO COMPUTERS AND BASIC 1
What Is BASIC? 1
Where Did BASIC Originate? 2
Where Is the Computer? 2
Terminals 4
How to Start 6

2 INTRODUCTION TO BASIC 9
Objectives 9
Getting On and Off the Computer 9
Requirements for BASIC Programs 9
Telling the Computer What to Do 9
Entering and Controlling Programs 9
Discovery Exercises 10
Getting On and Off the Computer 10
Computer Work 12
Discussion 20
Getting On and Off the Computer 20
Requirements for BASIC Programs 20
Telling the Computer What to Do 22
Practice Test 23
Entering and Controlling Programs 23

3 COMPUTER ARITHMETIC AND PROGRAM MANAGEMENT 27
Objectives 27
Arithmetic on the Computer 27
Parentheses ( ) in Computations 27
E Notation for Numbers 27
Variable Names in BASIC 28
Storing and Retrieving Programs 28
Discovery Exercises 28
Error Correction 28
Computer Work 30
Discussion 40
Arithmetic on the Computer 40

Parentheses in Computations 42
E Notation for Numbers 43
Variable Names in BASIC 44
Storing and Retrieving Programs 47
Practice Test 49

4 INPUT, OUTPUT, AND SIMPLE APPLICATIONS **53**
Objectives 53
Getting Numbers into a BASIC Program 53
Printing Out Variables and Strings 53
Spacing the Printout 53
The REMARK Statement 53
Simple Applications 54
Discovery Exercises 54
Computer Work 54
Discussion 62
Getting Numbers into a BASIC Program 62
Printing Out Variables and Strings 64
Spacing the Printout 65
The REMARK Statement 67
Program Examples 68
Example 1 Unit Prices 69
Example 2 Converting Temperature 70
Example 3 Sum and Product of Numbers 71
Problems 73
Practice Test 76

5 DECISIONS, BRANCHING, AND APPLICATIONS **79**
Objectives 80
Making Decisions in Programs 80
Program Applications 80
Finding Errors in Programs 80
Discovery Exercises 80
Computer Work 81
Discussion 84
Transfer without Conditions 84
Transfer on Conditions 85
Program Examples 88
Example 1 Printout of Number Patterns 89
Example 2 Automobile License Fees 90
Example 3 Averaging Numbers 94
Finding Errors in Programs 96
Translating BASIC Statements 96
Tracing BASIC Programs 97
Problems 103
Practice Test 105

6 LOOPING AND FUNCTIONS 107

Objectives 107

Built-in Looping 107

Built-in Functions 108

Program Applications 108

Discovery Exercises 108

Computer Work 108

Discussion 116

Built-in Looping 116

Built-in Functions 120

Program Examples 124

Example 1 Finding the Average of a Group of Numbers 124

Example 2 Temperature Conversion Table 125

Example 3 Exact Division 126

Example 4 Depreciation Schedule 127

Problems 130

Practice Test 133

7 WORKING WITH COLLECTIONS OF NUMBERS 135

Objectives 135

Single- and Double-Subscripted Variables 135

Matrix Commands 135

Program Applications 135

Discovery Exercises 135

Subscripts 136

Computer Work 138

Discussion 144

Single- and Double-Subscripted Variables 144

Saving Space for Arrays 147

Subscripted Variables and FOR NEXT Loops 148

MAT Commands in BASIC 149

Program Examples 153

Example 1 Examination Grades 153

Example 2 Course Grades 156

Example 3 Array Operations 158

Problems 160

Practice Test 163

8 "DO-IT-YOURSELF FUNCTIONS" AND SUBROUTINES 165

Objectives 165

"Do-It-Yourself Functions" 165

Subroutines 165

Program Applications 165

Discovery Exercises 166

Computer Work 166

Discussion 171

"Do-It-Yourself Functions" 172

Subroutines 173
Program Examples 177
Example 1 Rounding Off Dollar Values to Cents 177
Example 2 Carpet Estimating 178
Problems 182
Practice Test 185

9 RANDOM NUMBERS AND SIMULATIONS **187**
Objectives 187
Characteristics of Random-Number Generators 187
Random Numbers with Special Characteristics 187
Programming and Simulations 187
Discovery Exercises 188
Setting Up the Random-Number Generator 188
Computer Work 189
Discussion 192
Random-Number Generators 193
Designing Sets of Random Numbers 194
Troubleshooting Programs That Use Random Numbers 195
Program Examples 195
Example 1 Flipping Coins 195
Example 2 Random Integers 197
Example 3 Distribution of Random Numbers 197
Example 4 Birthday Pairs in a Crowd 198
Problems 200
Practice Test 200

APPENDIX A COMPUTER SYSTEM COMMANDS **203**
A-1 Computer List 203
A-2 Sign-on Instructions
A-3 Sign-off Instructions 204
A-4 Display the Program in Work Space 205
A-5 Execute the Program in Work Space 205
A-6 Clear Out the Program in Work Space 205
A-7 Jump Out of an Input Loop 206
A-8 Interrupt a Running Program 206
A-9 Delete a Character While Typing 206
A-10 Delete an Entire Line While Typing 206
A-11 Respond to System Error Messages 207
A-12 Move a Program from Memory to Work Space 207
A-13 Move a Program from Work Space to Memory 207
A-14 Clear Out a Program in Memory 208
A-15 Name a Program in Work Space 208
A-16 Display a Catalog of Programs in Account 208
A-17 Produce a Different Sequence of Random Numbers 208

A-18 Produce the Same Sequence of Random Numbers 209
A-19 Blank Form for Computers Not Listed in A-1 209

Practice Test Solutions 211

Solutions to Odd-numbered Problems 223

Index 243

1

Introduction to Computers and BASIC

Computers are now a common part of our lives. We may not see them, but they are there, involved in some way in most of our daily activities. Businesses of all sizes, educational institutions, various branches of government—none would be able to handle the bewildering quantity of information that seems to characterize our society except for computers. Only recently, however, has it become common for students at all levels to use computers routinely in their educational activities. As the price of computers continues to drop, this trend will surely continue. More and more people will need to know how to use computers if they are to participate fully in our society.

WHAT IS BASIC?

You are about to embark upon the study of a computer language called BASIC. BASIC is a very specialized language designed to permit you and the computer to understand and communicate with one another. This language is not complicated and is certainly much easier to learn than a spoken language such as Spanish or French. Even so, BASIC does have a simple vocabulary consisting of a few words, a grammatical structure, and rules of use just like any other language. Your main tasks will be to learn the vocabulary of BASIC, become used to its grammar rules, and begin to see how the language permits you to use the computer to do your bidding. The level of mathematics involved has intentionally been kept very low. Therefore if you feel a bit rusty in your mathematical skills, don't be too concerned. As we proceed through BASIC, you will have an opportunity to brush up on some elementary mathematics.

A very effective way to learn about anything is to observe details and characteristics while actually performing a task: the "discovery" method. This is the strategy that will be used in this book. You will be asked to begin each chapter with a session on a computer terminal. After following the directions and watching closely what the computer does in response to your instructions, you will begin to acquire a "feel" for BASIC. Once you have this type of understanding, you can proceed more profitably to study written material that summarizes what you have learned. Thus, the directed exercise on a computer terminal is a key part of learning about BASIC as presented in this book.

WHERE DID BASIC ORIGINATE?

The original version of BASIC was designed and written at Dartmouth College under the direction of Professors John G. Kemeny and Thomas E. Kurtz. In September 1963 work began on a project to perfect the concept of time sharing on a computer and to create a programming language written from the user's point of view. A very interesting sidelight was that much of the actual programming on the project was done by undergraduate students at Dartmouth. The birthday of BASIC is May 1, 1964, so the language is still a teen-ager.

The success of this pioneering effort at Dartmouth soon attracted national attention, and very quickly other institutions became interested. The rest is history. Today nearly every time-sharing computer supports the BASIC language. BASIC itself has grown significantly in both power and capability from its early versions. Each year the percentage of total computer activities done in BASIC increases. What started as a project at a single college is now an established part of the computer industry throughout the world.

WHERE IS THE COMPUTER?

Your contact with the computer will be through a computer terminal with a typewriter-like keyboard. You can send instructions and messages to the computer by typing on the keyboard. Likewise, the computer can send information to you by typing out on your terminal. It makes very little difference where the computer itself is. It could be right beside your terminal (or, what with the astonishing decrease in size, *inside* your terminal!), in the next room, or halfway around the world. All that is necessary is that messages can flow back and forth between the computer and your terminal. These messages can pass over wires, through phone circuits, via radio transmission, or even by transmission of signals via satellites in orbit.

As you look around the terminal room, you will more than likely notice other people working at terminals. Each terminal is not connected to its own private computer, as this could be quite expensive. Instead, the terminals are usually connected to the same computer. The computer handles all the terminals, giving its attention briefly to each of the users and moving quickly from one terminal to the next. This happens so rapidly that you may feel you have the entire computer to yourself. Machines like this are known as *time-*

sharing computers and can handle from a few terminals to several hundred at the same time, depending on size and cost (see Figures 1 and 2.)

Quite often, introductory computer programming texts spend a considerable amount of time on the hidden workings and inner mechanisms of computers. This will *not* be done in this book. As a typical user, you are most likely interested in learning how to use the computer rather than the details of how it works. In precisely the same sense, you do *not* need to be an expert in internal combustion engines to drive an automobile! Consequently we will pursue one, and only one, goal throughout the book: how to write and execute programs in BASIC.

Even though we will not become involved with the specifics of computer construction, we should pause to note the astonishing changes that have taken place in computers in the past ten years. What was once a giant power-gobbling electronic monster can now be placed on a tiny chip. We have just begun to see the changes in our lives that this sophisticated technology will

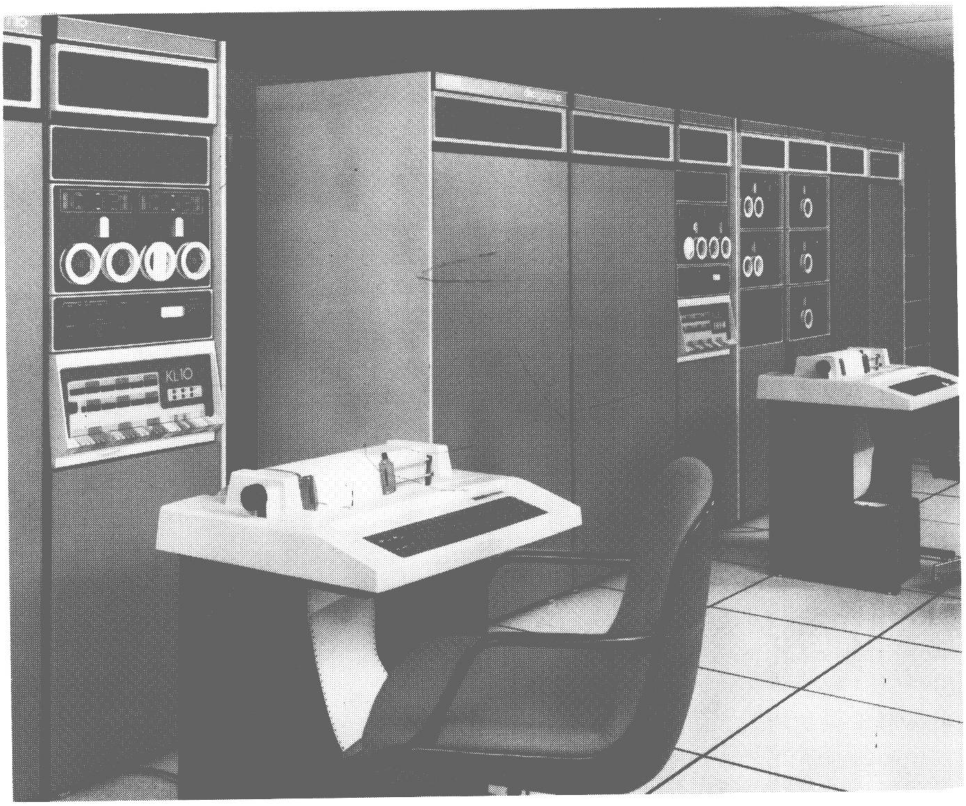

Figure 1 **Large time-sharing computer, DECsystem-10. (*Copyright © 1976 by Digital Equipment Corp. All rights reserved.*)**

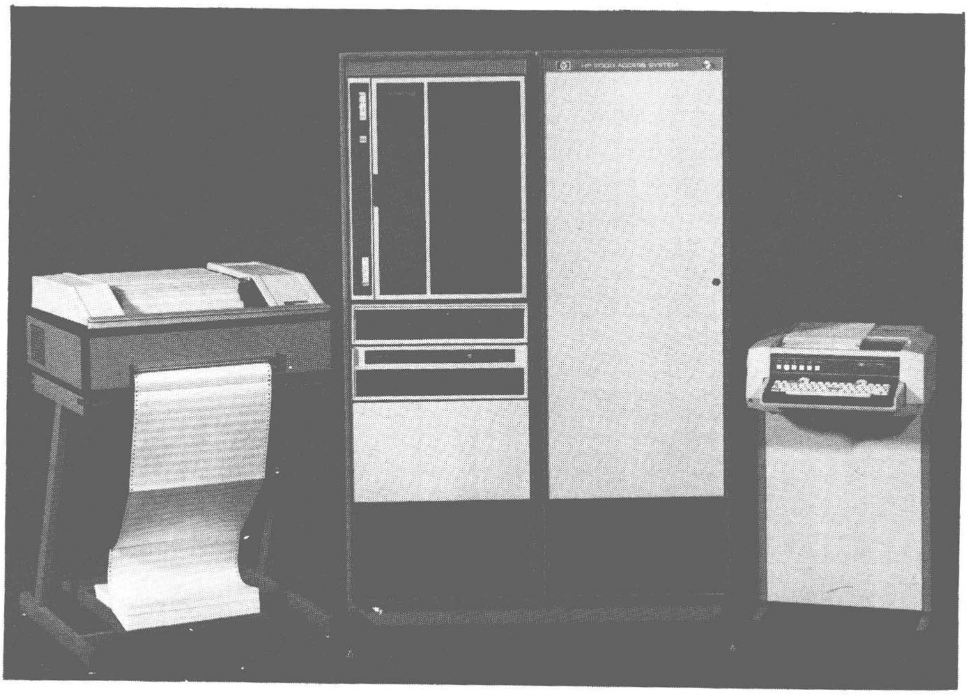

Figure 2 Small time-sharing computer, Hewlett Packard 2000 System. ***(Courtesy of Hewlett Packard Co.)***

bring about. An entire industry exists today to build and distribute the marvelous pocket calculators that are so common but were unknown just a few years ago. Within a few years, small computers on chips will be used in automobiles to monitor and control various functions (see Figure 3). Probably the one thing that we can count on is that computers of all sizes and with many different characteristics will become more and more common.

TERMINALS

There are three different kinds of computer terminals in general use today. The first is the printing (or *hard-copy*) terminal, which prints the results on paper. Probably the Teletype terminal (Figure 4) is the most common of these, although newer types of hard-copy terminals are coming into general use. The advantage of a printing terminal is that you have a record of your work with the computer which can be saved for later use (see Figure 5).

A second type of terminal is the CRT (which stands for *cathode ray tube*) terminal (Figure 6). Instead of a printed copy on paper, this terminal shows the output on a television-like screen (the cathode ray tube). Sometimes it is possible for you to use a separate machine to make a paper copy of whatever appears on the face of your CRT terminal.

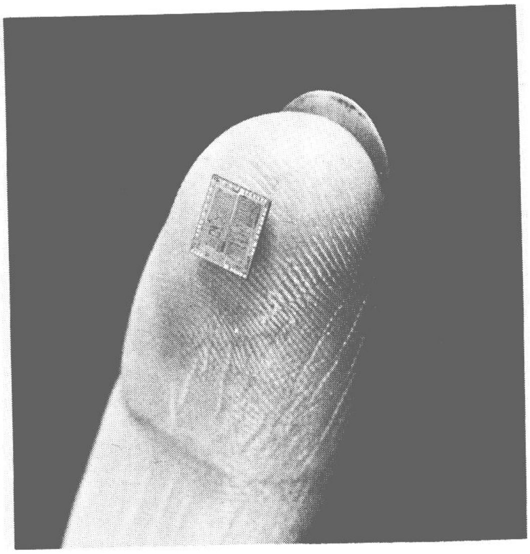

Figure 3 Computer chip on a finger. (*Photograph courtesy of Intel.*)

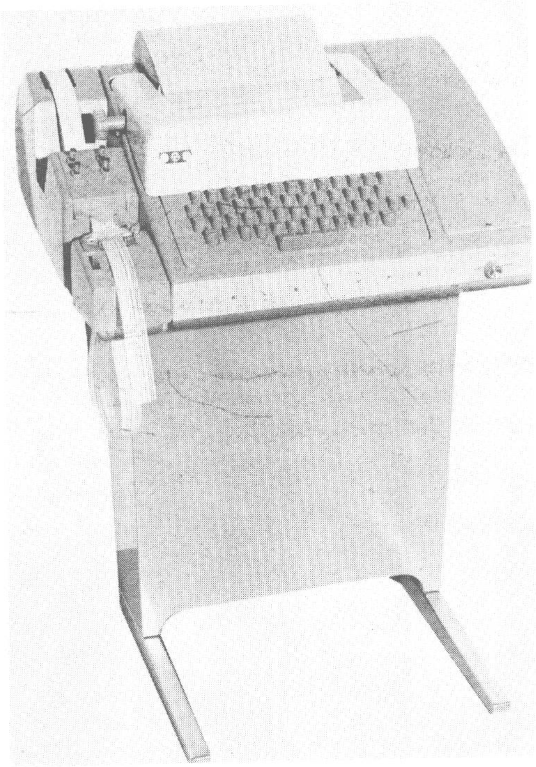

Figure 4 Model 33 teletype terminal. (*Photograph courtesy of Teletype Corp.*)

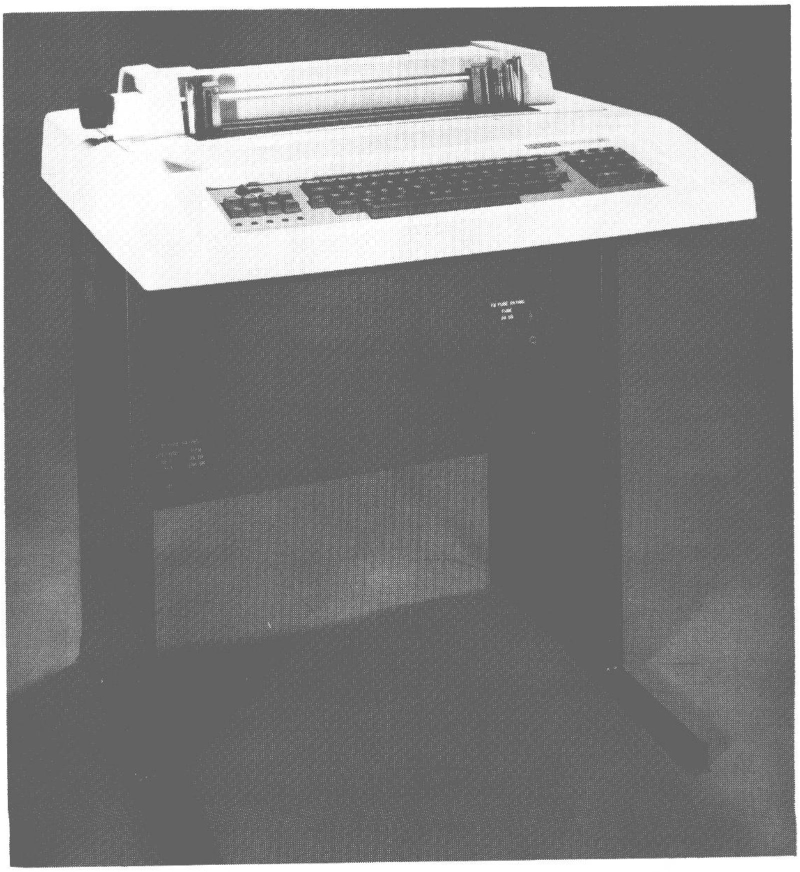

Figure 5 DECwriter II hard-copy terminal. ***(Copyright © 1976 by Digital Equipment Corp. All rights reserved.)***

Finally, there are *graphic* terminals. These look like ordinary CRT terminals but can also draw lines to produce pictures, charts, and graphs. It is this ability that leads to the name "graphic" terminal. Since we will not become involved with graphics in this book, if you find a graphic terminal use it as a CRT terminal and we will ignore the graphics ability.

The specific details of use may vary slightly from terminal to terminal. The differences are slight, however, and you should have little difficulty in adapting to the different types that may be found.

HOW TO START

You should approach each chapter in the book in the same way. The material has been organized with special learning patterns in mind, and any changes will be less effective and require more of your time.