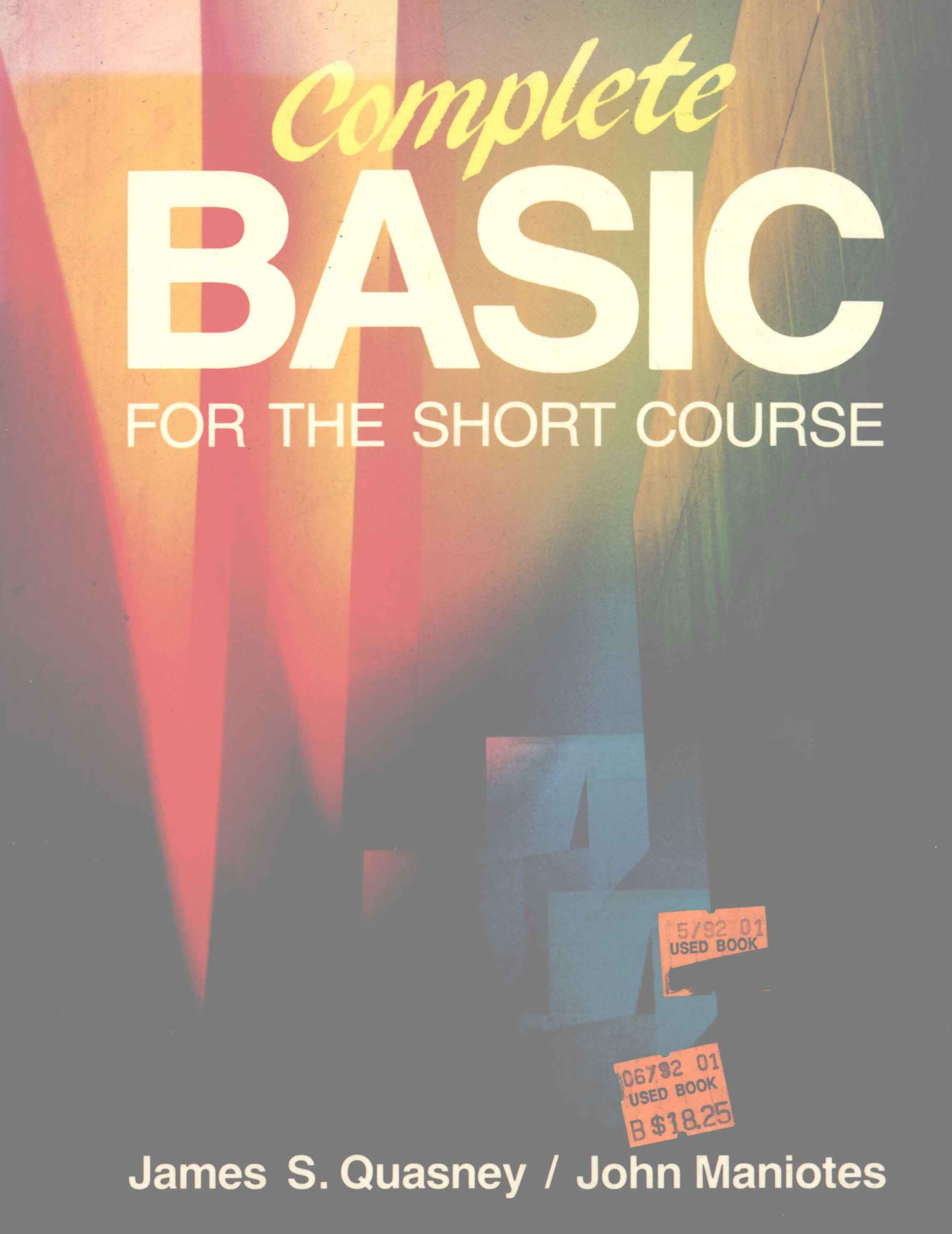
# *Complete*
# BASIC
## FOR THE SHORT COURSE





James S. Quasney / John Maniotes

# *Complete* BASIC

## FOR THE SHORT COURSE

**James S. Quasney**

**John Maniotes**

***Purdue University Calumet***

**Boyd & Fraser Publishing Company**

**Boston**

*To our wives: Linda and Mary*
*The Quasney tribe: Lisa, Jeff, Marci, Jodi, Amanda and Nikole*
*The Maniotes clan: Dionne, Sam and Andrew*

Editor: Tom Walker
Production manager: Erek Smith
Development editor: Sharon Cogdill
Associate editor: Donna Villanucci
Production editor: Toni Rosenberg
Design: Rafael Millán
Cover: Carol Rose



Manufactured in the United States of America

**Library of Congress Cataloging in Publication Data**

Quasney, James S.
Complete BASIC.
Includes index.
1. Basic (Computer program language) I. Maniotes, John. II. Title.
QA76.73.B3Q373 1985 001.64'24 84-29298
ISBN 0-87835-151-5

3 4 5 6 7 SZ 3 2 1 0 9

# *Complete*
# BASIC
## FOR THE SHORT COURSE

**The Boyd & Fraser Programming Language Series includes these fine titles:**

BASIC Fundamentals and Style
Beginning Structured COBOL
Advanced Structured COBOL: Batch and Interactive
Comprehensive Structured COBOL
Fortran 77 Fundamentals and Style
Pascal

**Other outstanding programming language texts from Boyd & Fraser include:**

BASIC Programming with Structure and Style
Complete BASIC: For the Short Course
Standard BASIC Programming: For Business and Management Applications
Structured Fortran 77 Programming
Structured Fortran 77 Programming: For Hewlett-Packard Computers
Pascal Programming: A Spiral Approach

**Also available from Boyd & Fraser are these quality texts:**

Data Communications Software Design
The Business Computing Primer: Using Microcomputer Software
Computers and Man

Cover Photo: David S. Hughes, Stock/Boston

# PREFACE

## THE BASIC PROGRAMMING LANGUAGE

People in almost every walk of life will come into contact with computers and BASIC, the most popular language used to program them. BASIC has emerged as the most widely used programming language in the world. BASIC has gained strong support from the manufacturers of personal computer and timesharing systems because it is easy to learn, powerful and flexible.

## ABOUT THIS BOOK

*Complete BASIC: For the Short Course* is ideally suited for use in a short introductory programming course. The text was specifically designed, however, to meet the needs of instructors teaching an Introductory Data Processing course in which BASIC programming is a component. *Complete BASIC: For the Short Course* is a perfect complement to such texts as: Capron and Williams, COMPUTERS AND DATA PROCESSING, Second Edition (Benjamin-Cummings); Kroenke, BUSINESS COMPUTER SYSTEMS: AN INTRODUCTION, Second Edition (Mitchell); Long, INTRODUCTION TO COMPUTERS AND INFORMATION PROCESSING (Prentice-Hall); Mandell, COMPUTERS AND DATA PROCESSING TODAY (West); Sanders, COMPUTERS TODAY (McGraw-Hill); Shelly and Cashman, COMPUTER FUNDAMENTALS: FOR AN INFORMATION AGE (Anaheim); Stern and Stern, COMPUTERS AND SOCIETY (Prentice-Hall); and similar texts.

## OBJECTIVES OF THIS BOOK

The book was developed with the following objectives:

1. to instill good problem-solving habits
2. to acquaint the reader with the syntax of the BASIC programming language
3. to introduce the reader to the proper and correct way to design and write programs
4. to encourage independent study and help those working alone on their own systems

## LEVEL OF INSTRUCTION

No previous experience with a computer is assumed, and no mathematics beyond the freshman high-school level is required. The book is specifically written for the student with average ability, for whom continuity, simplicity and practicality are essential.

## DISTINGUISHING FEATURES

The distinguishing features of this book include the following:

### Problem-oriented Approach

Over 100 BASIC programs plus many partial programs, representing a wide range of practical applications, are used to introduce specific statements and the proper and correct way to write programs.

### Emphasis on the Program Development Cycle

The program development cycle is presented early in Chapter 1 and is used throughout the book. Good design habits are reinforced, and special attention is given to testing the design before attempting to implement the logic in a program.

### Machine Specificity

Various dialects of BASIC are consistently highlighted throughout the book, which accentuates microsoft BASIC. General Forms or footnotes are used to indicate the availability and form of BASIC statements for the following computer systems: Apple, COMMODORE, DEC Rainbow/MBASIC-86, DEC VAX-11, IBM PC/MICROSOFT, Macintosh/MICROSOFT and TRS-80 Model 4. Other requirements that differ among systems, such as how to get a hard copy, clearing the screen, the composition of variable names and precision, are presented in tables.

### Flowcharts Illustrating Program Design

Flowcharting is an excellent pedagogical aid and one of the tools of an analyst or programmer. Many of the program examples include program flowcharts of the logic to demonstrate programming style, design and documentation considerations. Line numbers have been placed on the top left corner of the symbols to better illustrate the relationship between the flowchart and the program.

### BASIC at Work

The book contains 13 completely solved and annotated actual case studies, illuminating the use of BASIC and computer programming in the real world. Emphasis is placed on problem analysis, program design and an in-depth discussion of the program solution.

### Programming Exercises with Sample Input and Output

Over 30 challenging Programming Exercises with sample Input and Output are included at the end of the chapters. These problems are in order from most simple to most difficult. All the problems include sample input data and the corresponding output results. Solutions to these exercises are given in the *Instructor's Manual and Answer Book* and are also available to instructors from our publisher on an IBM PC-compatible diskette.

### Structured Programming Approach

Particular attention is given to designing proper programs using the three logic structures of structured programming: Sequence, Selection (If-Then-Else and Case)

and Repetition (Do-While and Do-Until). Consistent use of the IF–THEN–ELSE statement, logical operators and the WHILE and WEND statements helps minimize dependence on the GOTO statement.

### Complete Coverage of Sequential File Processing

Complete coverage of sequential files provides students with knowledge central to a real programming environment. Topics include creating sequential files and writing reports to auxiliary storage.

### Interactive Applications (Menu-Driven Programs)

Although batch processing is discussed in detail, the primary emphasis is on interactive processing. The INPUT, PRINT and Clear Screen statements are introduced early in Chapter 2. Menu-driven programs are illustrated to familiarize students with the type of programming proliferating in today's world.

### Debugging Techniques

Characteristic of a good programmer is confidence that a program will work the first time it is executed. This confidence implies that careful attention has been given to the design and that the design has been fully tested. Still, errors do occur, and they must be corrected. Throughout the book, especially in Appendix A, efficient methods for locating and correcting errors are introduced. Both TRON and TROFF, as well as other techniques, are discussed in detail.

### Concise Introduction to Computers

A concise discussion of how computers operate is included at the beginning of the book. Equal attention is given to personal computers and timesharing.

### Use of Second Color

The effective use of a second color throughout the book enhances readability, highlights key concepts and facilitates easy reference. This is especially important for readers using this book in a self-study environment or as a reference.

### Solutions to Programming Exercises on IBM PC Diskette

An IBM PC-compatible diskette with all the program solutions to the 30 Programming Exercises is available upon request from our publisher, Boyd & Fraser, for instructors who adopt this book.

## INSTRUCTOR'S MANUAL AND ANSWER BOOK

The *Instructor's Manual and Answer Book*, available to instructors upon request from Boyd & Fraser, includes transparency masters from each chapter of the text, chapter-by-chapter objectives and vocabulary lists, lecture outlines, program solutions to all 30 programming assignments in the book, answers to the odd-numbered BASIC Self-Text Exercises, and a test bank that includes true/false, short-answer, fill-in and multiple-choice questions for quizzes and tests.

**ACKNOWLEDGMENTS** We were fortunate in having a group of reviewers whose critical evaluations of earlier drafts of the manuscript were of great value. Special thanks go to Professors Chester Bogosta of Saint Leo College, John J. Couture of San Diego City College, Syed Shahabuddin of Central Michigan University, Sumit Sircar of the University of Texas at Arlington, Marilyn Markowicz of Purdue University Calumet and Mick L. Watterson of Drake University. No book is possible without the motivation and support of an editorial staff. Therefore, our final acknowledgment and greatest appreciation is reserved for the following at Boyd & Fraser: to Tom Walker, editor in chief, for the opportunity to write this book and his constant encouragement; to Sharon Cogdill, a talented development editor, whose creative ideas and organizational abilities are visible throughout the book; to Toni Rosenberg, the production editor, who did a marvelous job editing the book under a tight production schedule, and to Donna Villanucci, associate editor, who was an invaluable asset during the writing and production of this book as well as of the accompanying *Instructor's Manual and Answer Book.*

Hammond, Indiana
January 1985

James S. Quasney
John Maniotes

# CONTENTS

# 1 COMPUTERS AND PROBLEM-SOLVING: AN INTRODUCTION

## 1.1 WHAT IS A COMPUTER?

A **computer** is a machine that can accept data, process the data at high speeds, and give the results of these processes in an acceptable form. A more formal definition of a computer is given by the American National Standards Institute (ANSI), which defines a computer as a device that can perform substantial computation, including numerous arithmetic or logic operations, without intervention by a human operator.

### Advantages of a Computer

The major advantages of a computer are its speed, its accuracy, and its ability to store and have ready for immediate recall vast amounts of data. Modern computers can also accept data from anywhere via telephone lines or satellite communications. They can generate usable output, like reports, paychecks, and invoices, at several thousand lines per minute.

Computers can handle large amounts of data and tedious and time consuming work without ever tiring, which makes them indispensable for most businesses. In fact, computers have been among the most important forces in the modernization of business, industry and society since World War II. Keep in mind, however, that with all their capabilities, computers are not built to think or reason. They extend our intellect, but they do not replace thinking.

## 1.2 COMPUTER HARDWARE

**Computer hardware** is the physical equipment of a computer system. The equipment may consist of any combination of mechanical, magnetic, optical, electrical and electronic devices. Although many computers have been built in different sizes, speeds, and costs, and with different internal operations, most of them have the same basic subsystems (see Figures 1.1, 1.2 and 1.3).

### Input

An **input unit** is a device that allows **programs** (instructions to the computer) and **data** (like rate of pay, hours worked, and number of dependents) to enter the computer system. This device converts the incoming data into electrical impulses which are sent to the other units of the computer. A computer system usually has a **keyboard** for input. Other common input devices include a joystick, mouse, and floppy diskette unit.

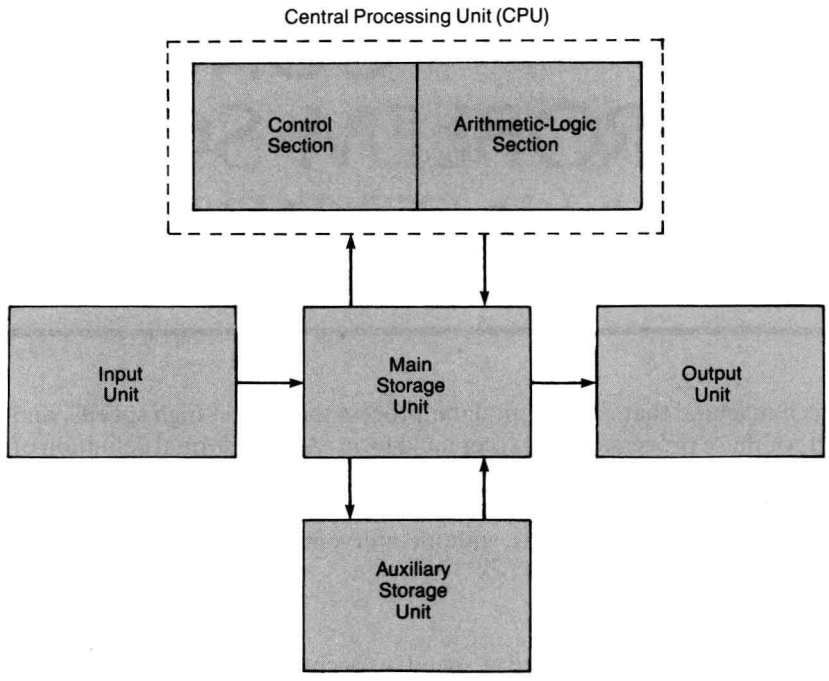



**Figure 1.1 Basic structure of a digital computer, where the arrows represent the flow of data.**

**Figure 1.2 The IBM Personal Computer (Courtesy IBM).**

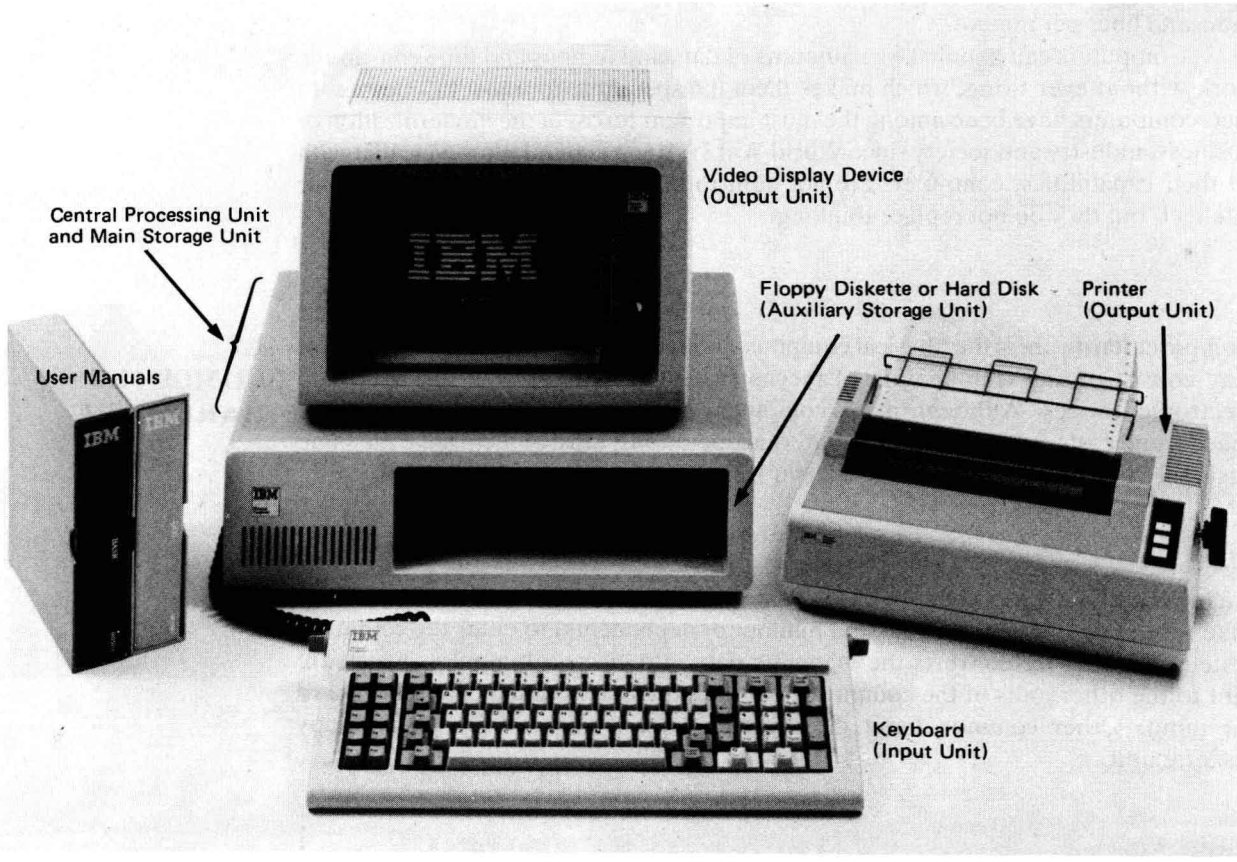



**Figure 1.3 A DEC VAX-11/780 medium-scale computer system with interactive processing capabilities (Courtesy Digital Equipment Corp.).**

## Main Storage

After the instructions and data have entered the computer through an input unit, they are stored in the computer's **main storage** unit. Since computers can process vast amounts of data in a short time and since some can perform millions of calculations in just one second, the storage unit must be able to retain large amounts of data and make any single item rapidly available for processing.

Main storage in a computer is divided into locations, each having an **address**. When instructions and data are entered, they are stored in various locations of main storage. The computer leaves data in a storage location until it is instructed to replace it with new data. While a data item is in storage, the computer can "look it up" as often as it is needed. Thus, when data is retrieved from a storage location, the stored contents remain unaltered. When you instruct the computer to put new data in that location, it replaces old data.

## Central Processing Unit (CPU)

The **CPU** controls and supervises the entire computer system and performs the actual arithmetic and logic operations on data as specified by the written program. The CPU is divided into the **arithmetic-logic section** and the **control section** (see Figure 1.1).

The arithmetic-logic section performs such operations as addition, subtraction, multiplication, division, transferring, storing, and setting the algebraic sign of the

results. Depending on the cost and storage capacity of the computer, the speed of the arithmetic unit will range from several thousand to many millions of operations per second.

The arithmetic-logic section also carries out the decision-making operations required to change the sequence of instruction execution. These operations include testing various conditions, such as deciding the algebraic sign of a number or comparing two characters for equality. The result of these tests causes the computer to take one of two or more alternate paths through the program.

The control section directs and coordinates the entire computer system according to the program developed by the programmer and placed in main storage. Its primary function is to analyze and initiate the execution of instructions. This means that the control section has control over all other subsystems in the computer system. It can control the input of data and output of information and routing of data and information between auxiliary storage and main storage or between main storage and the arithmetic-logic section.

## Auxiliary Storage

The function of the **auxiliary storage** unit is to store data and programs that are to be used over and over again. Common auxiliary storage devices are the **magnetic tape, hard disk** and **floppy diskette.**

Both magnetic tape and disk can be used to store programs and data for as long as desired. A new program entering the system erases the previous program and data in main storage, but the previous program and data may be permanently stored on an auxiliary storage device for recall by the computer.

In a business, files containing employee records, customer records, accounts receivable or payable data, and inventory data are stored on magnetic tape or disk. Programs written to print paychecks, invoices and management reports are also stored on these auxiliary storage devices. Without auxiliary storage, all programs and data would have to be manually entered through an input device every time an application was processed.

## Output

When instructed by a program, the computer can communicate the results of a program to **output units.** A computer usually has a video display device for output. Other common output devices include a printer, plotter, and floppy diskette unit.

The **video display device**, also called a **CRT** or **VDT**, is similar to the tube in a television set and can be used to display the output results in the form of words, numbers, graphs, or drawings (see Figures 1.2 and 1.3).

High-speed line printers, some of which can operate at speeds of more than 2,000 lines per minute, can prepare invoices, checks, report cards, and other output. If the results of a program are to be processed further, the information can also be placed on magnetic disk, diskette or tape. The information can be used later as data for the next problem or sent over telephone lines to another computer for further processing.

## 1.3 THE STORED PROGRAM CONCEPT

In order for a computer to take action and produce a desired result, it must have a step-by-step description of the task to be accomplished. The step-by-step description is a series of precise instructions called a **program.** When these instructions are placed into the main storage unit of a computer, they are called the **stored program**. Main storage not only stores data but also the instructions which tell the computer what to do with the data. The stored program gives computers a great deal of flexibility. Without it the computer's ability to handle tasks would be reduced to that of a desk calculator.

Once the program is stored, the first instruction is located and sent to the control section, where it is interpreted and executed. Then the next instruction is located, sent to the control section, interpreted and executed. This process continues automatically, instruction by instruction, until the program is completed or until the computer is instructed to halt.

In order for the computer to perform still another job, a new program must be stored in main storage. Hence, a computer can be easily used to process a large number of different jobs.

## 1.4 COMPUTER SOFTWARE

**Computer software** is a set of programming languages and programs concerned with the operation of a computer system.

Programming languages are classified as **low-level languages** (like machine language and assembly language) and **high-level languages** (like Ada, BASIC, Pascal, COBOL, FORTRAN and PL/I). Early generation computers required programmers to program in **machine language**, and this language was different for each computer manufacturer's system.

Currently, most applications are programmed in one of the many popular high-level languages listed in Table 1.1. A high-level language is generally machine or computer independent; this means that programs written in a high-level language like BASIC are portable—they can be transferred from one computer system to another with little or no change in the programs.

## BASIC at Work

**Computing an Average**

Program 1.1 illustrates a program written in BASIC. It instructs the computer to compute the average of three numbers, 17, 23, and 50.

**Program 1.1**

BASIC Program

```
100 REM PROGRAM 1.1
110 REM COMPUTING AN AVERAGE
120 REM ********************
130 LET A = (17 + 23 + 50)/3
140 PRINT "THE AVERAGE IS"; A
150 END
```

System Command

```
RUN
```

Displayed Result

```
THE AVERAGE IS 30
```

The displayed answer, found below the word RUN, is 30. Even though we are deferring detailed explanations about this program until the next chapter, Program 1.1 gives you some indication of instructing a computer to calculate a desired result using the BASIC language.