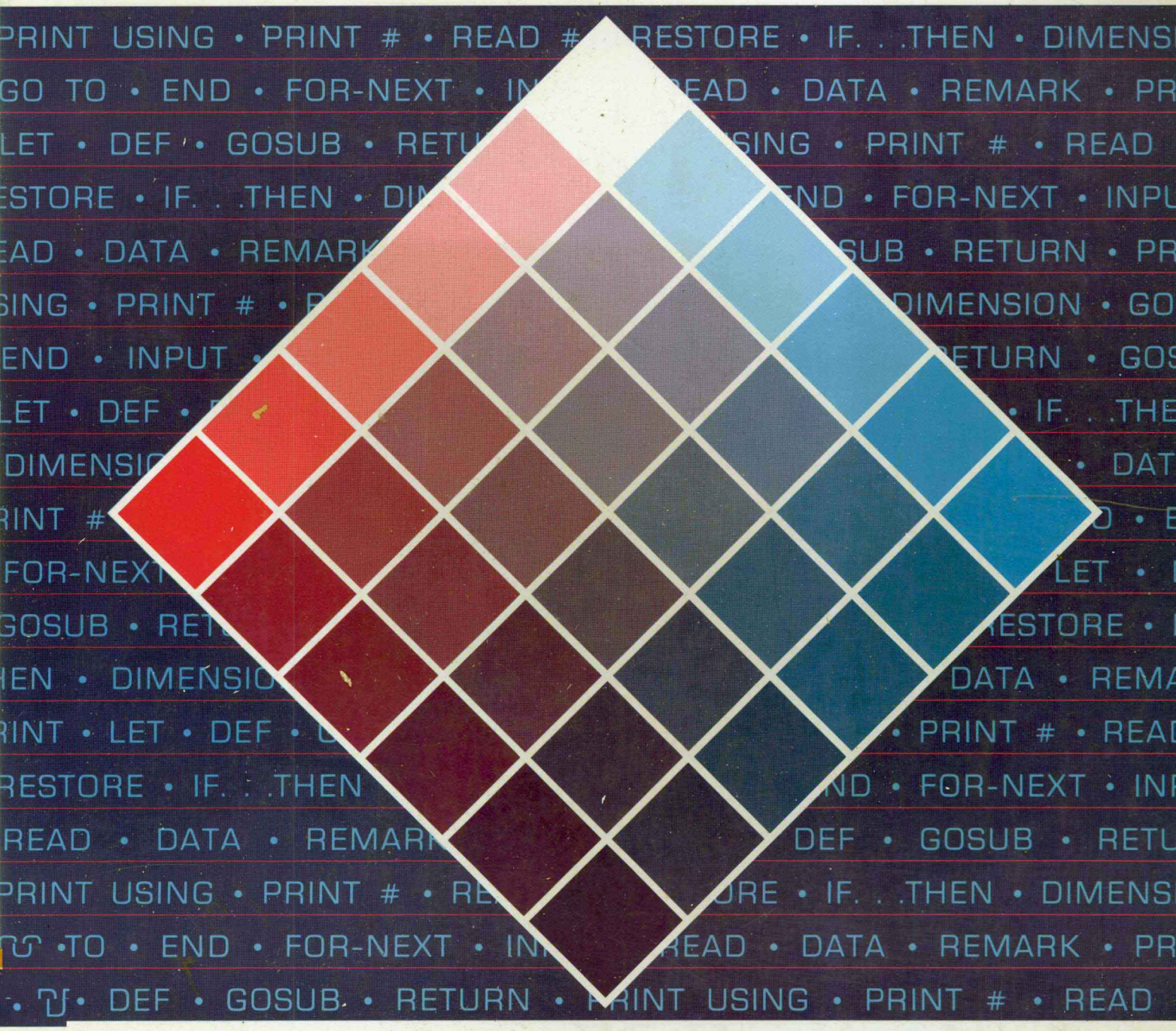
PROGRAMMING in BASIC

Communicating with Computers

Jeffrey Frates



PROGRAMMING in BASIC

Communicating with Computers

Jeffrey Frates

Los Medanos College

Prentice-Hall, Inc., *Englewood Cliffs, New Jersey 07632*

Library of Congress Cataloging in Publication Data

FRATES, JEFFREY.
Programming in BASIC.
Includes index.
1. Basic (Computer program language) 2. Microcomputers—Programming. I. Title
QA76.73.B3F73 1985 001.64'24 84-11610
ISBN 0-13-729369-0

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

Editorial/production supervision
and interior design: Margaret McAbee and Joan L. Stone
Cover design: Ken Alexander
Manufacturing buyer: Ed O'Dougherty

Illustrations from Jeffrey Frates and William Moldrup, *Introduction to the Computer: An Integrative Approach* 2nd Ed., © 1984, pp. 239, 241, 242, 243, 244, 245, and 246. Reprinted by permission of Prentice-Hall, Inc., Englewood Cliffs, N.J. Illustrations from Jeffrey Frates and William Moldrup, *Computers and Life: An Integrative Approach*, © 1983, pp. 24, 53, 54, and 57. Reprinted by permission of Prentice-Hall, Inc., Englewood Cliffs, N.J.

Figure 1-2 courtesy of INTEL Corp. Figures 1-3 and 1-4 courtesy of IBM. Figure 1-5 courtesy of Prime. Figure 1-6 courtesy of Radio Shack.

ISBN 0-13-729369-0 01

PRENTICE-HALL INTERNATIONAL, INC., *London*
PRENTICE-HALL OF AUSTRALIA PTY. LIMITED, *Sydney*
EDITORA PRENTICE-HALL DO BRASIL, LTDA., *Rio de Janeiro*
PRENTICE-HALL CANADA INC., *Toronto*
PRENTICE-HALL OF INDIA PRIVATE LIMITED, *New Delhi*
PRENTICE-HALL OF JAPAN, INC., *Tokyo*
PRENTICE-HALL OF SOUTHEAST ASIA PTE. LTD., *Singapore*
WHITEHALL BOOKS LIMITED, *Wellington, New Zealand*

To the Instructor

BASIC programming has always presented difficulties for instructors. So many different forms of the language exist, employing many different computer systems, that it is very difficult to teach students one "right" way to program in BASIC. No text can cover every aspect of the language as it is implemented on every different computer, and those that attempt to do so often confuse both the students and the instructor. To teach BASIC as well as learn it, one must focus on the general structure of the language and the important concepts that are elementary in understanding this programming language.

Further, instructors are often faced with students who are not familiar with effective problem-solving or logical analysis skills, or who are apprehensive about the use of mathematics in programming. Because BASIC is one of the simpler programming languages for students to learn, it is here that the vast majority of students wishing to learn something about the computer and its capabilities must receive a solid foundation in programming skills.

This text is designed to be used in either of two instructional settings. First, it may be used as a supplement to an Introduction to Computers, Computers and Society, or Computer Literacy course, where the primary text does not provide enough depth of actual programming material. Second, it may be used as a stand-alone text for an Introduction to BASIC Programming course, in which the goal is to expose a wide cross section of students to elementary programming concepts using BASIC.

This text does not profess to teach comprehensive BASIC; it does not attempt to cover how BASIC is implemented in all of its different versions. What the text does do is focus on the important standard BASIC concepts that are necessary for a solid foundation in understanding programming. Early in the book students are exposed to effective problem-solving methods. And

with these problem-solving methods students are taught programming style and techniques that can increase the clarity and effectiveness of programs that they may write.

Each programming chapter ends with a solved problem. A problem has been chosen for this section of the text (Integrating the Old and the New) that evolves throughout the book. As new techniques are introduced in each chapter, the problem at the end of the previous chapter is built upon, using the new programming techniques introduced in the current chapter, to form a more comprehensive program. This section first describes the nature of the problem, then the methods used to accomplish the objectives in the problem, then the actual solution to the problem in program form, and finally an analysis of how the program works, with specific references to sections or instructions within the program. This technique should enable students to evaluate the use of various new programming techniques in light of both current and past programming material presented.

This text assumes no prior knowledge of computers or programming. It is written to provide nonmathematically oriented students with a solid understanding of elementary BASIC programming concepts and techniques. One of the important features of this text is its chapter structure. Each of the chapters dealing with programming material is structured identically, which the author feels is important for students to become comfortable with the methods and know what to expect in each chapter presentation. Each chapter begins with a brief review of the material presented in the previous chapter, a preview of the material to be presented in the current chapter, and how the two are related or used to enhance one another.

The first two chapters of the text introduce students to fundamental understanding focusing on computer hardware, systems, software, and problem-solving analysis. Chapter 3 actually begins the programming material. From this point each chapter follows the same form and structure. The structure of each programming chapter is listed below.

* TRANSITION—briefly describes materials presented in earlier chapters and mentions the limitations of these techniques.
* NEW RESOURCES—introduces the new topics presented in the current chapter and explains how these new techniques improve or build upon materials presented in earlier chapters.
* BEWARE OF . . .—points out common errors and difficulties that may occur in using the programming techniques presented in the chapter.
* INTEGRATING THE OLD AND THE NEW—presents a problem to be solved using the programming techniques in the chapter. The problem at the end of each chapter is based on the problem at the end of the previous chapter but is enhanced to include the programming techniques presented in the current chapter. After each of these programs are several blank lines to be used for notes regarding aspects of the program that are unique to the user's particular computer or version of BASIC.
* EXERCISES—provides an extensive group of programming exercises that are based on material presented in the chapter. The ex-

ercises vary in difficulty, providing a challenge for students of every level of skill.

There are three appendices at the end of the book. Appendix A is a guide to entering, editing, and executing BASIC programs on several popular brands of microcomputers, and a typical programming session on a timeshare-oriented computer.

The second appendix focuses on problem solving. In the text the second chapter discusses methods for developing the solution to a programming problem. And in appendix B, these methods are applied to a specific problem, allowing students to follow, in narrative style, the development of a solution to a programming problem. This section carries students through each step in the programming process, including program definition, algorithm development, and flowchart design. Finally, the program that actually solves the stated problem is presented.

Appendix C is a language reference guide, presenting the statements available in BASIC, with a brief statement about their use.

As their instructor, you will wish to provide your students with specific programming instructions regarding the process on the system which they will be using in your computer facility. Always point out any differences in the way that BASIC instructions can be implemented on your computer equipment as opposed to how they are presented in the book. Good luck!

To the Student

As we know, computers have become a powerful force in our world. Until recently, they have been perceived as the tools of scientists, engineers, and mathematicians—too technical and complex for the average person to understand, let alone operate. We have come to a time, however, at which some understanding of the workings of a computer system is important for everyone, regardless of background, mathematical abilities, or occupational goal.

Over the years computers have become more powerful, yet easier to operate and understand. Two of the factors which have attributed to this are the advent of low cost microcomputers and timeshare computers as well as the development of programming languages that are relatively easy for nontechnically oriented students to master. Such a language is BASIC, which is the topic of this book. This book assumes no prior knowledge of computers, programming, or mathematics. It can be used as a supplement to an Introduction to Computers text, as a means of delving deeper into BASIC programming than the material presented in the primary text, or it can be used as a stand-alone text to teach fundamental concepts of programming in BASIC.

The first two chapters of the text focus on general computer concepts. The first chapter describes basic computer terminology, what a computer system is and how it operates, the different types of computer systems, and the role of programming in the use of computers. The second chapter focuses more specifically on the programming process. In that chapter the methods for effective program development are discussed, as well as suggestions for programming style to increase the effectiveness and understanding of the programs that you write.

The remaining chapters in the book are devoted to actual programming material. Each of these chapters has an identical structure, which enables

you to anticipate what you will be doing next and to continuously relate current materials with what you have learned in previous chapters. A brief description of the structure of each of these chapters is listed below.

* TRANSITION—briefly describes materials presented in earlier chapters and mentions the limitations of these techniques.
* NEW RESOURCES—introduces the new topics presented in the current chapter and explains how these new techniques improve or build upon materials presented in earlier chapters.
* BEWARE OF . . .—points out the common errors and difficulties that may occur in using the programming techniques presented in the chapter.
* INTEGRATING THE OLD AND THE NEW—presents a problem to be solved using the programming techniques in the chapter. The problem at the end of each chapter is based on the problem at the end of the previous chapter but is enhanced to include the programming techniques presented in the current chapter. After each of these programs are several blank lines to be used to make brief notes regarding aspects of the program that may be unique to your computer or version of BASIC.
* EXERCISES—provides an extensive group of programming exercises that are based on material presented in the chapter. The exercises vary in difficulty, providing a challenge for students of every level of skill.

Three appendices conclude this text. The first describes basic operating procedures on several popular computer systems. The second presents a programming problem, to which the solution is developed via the problem-solving methods presented in Chapter 2. The third appendix is a language reference, briefly listing and describing the instructions that are used in BASIC programs.

Your instructor will be your guide through this book. Let me stress, however, the importance of the use of several features of this book. Pay close attention to the problem-solving and programming style material in Chapter 2. Follow these procedures closely and use the solved problem in Appendix B as a guide to solve problems you may assigned or the exercises at the end of each chapter. Study closely the continuing problem presented at the end of each chapter. This will provide you with a better understanding of the relationship of the various programming techniques explained in each chapter. And finally, when you are unsure about the workings of a specific example presented in the book, enter that program into the computer that you are working with and attempt to make it run. In many cases, seeing a program with a particular technique in it actually running on a computer can help you understand how that particular concept works. Good luck!

Contents

TO THE INSTRUCTOR, xi

TO THE STUDENT, xv

1

INTRODUCTION TO COMPUTERS, 1

Introduction, *1*
The Electronic Computer, *2*
Computer Hardware, *4*
Input, *4* Central Processor and Data Storage, *4* Output, *6*
Computer Programming: Software, *7*
Computer Systems, *8*
Batch Computers, *8* Timeshare-Interactive Computers, *10* Microcomputers, *11*

2

PROGRAMMING METHODS AND STYLE, 15

Introduction, *15*
Programming Languages, *16*
Instructions, *16* Types of Languages, *16* Language Characteristics, *17*

Programming Methods, *17*
The Programming Process, *18* Defining the Problem, *19*
Plan the Solution, *20* Flowcharting, *20* Code the Solution, *23*
Testing the Solution, *24* Program Documentation, *24*
Programming Style, *25*
Program Organization, *25* Program Documentation, *28*
Interactive Programming, *30*
Summary, *31*

3

GETTING STARTED, 33

Transition, *33*
New Resources, *33*
Computer Programming, *33* Program Structure, *34* Statements, *34*
BASIC Keywords, *35*
REMARK Statement, *35* PRINT Statement, *35* END Statement, *36*
A Complete Program, *36* Numerical Values, *36* LET Statement, *36*
Numeric Variables, *38* Rules for Variable Names, *38*
Assigning Values to Variables, *39*
Beware of . . . , *40*
Integrating the Old and the New, *41*
Exercises, *42*

4

CALCULATIONS, 45

Transition, *45*
New Resources, *45*
Advanced Input, *45* READ and DATA Statement, *46* Calculations, *47*
Arithmetic Operations, *47* Priority of Operations, *49* More Printing, *50*
Print Zones, *50* SPACE Function, *51*
Beware of . . . , *53*
Integrating the Old and the New, *54*
Exercises, *56*

5

LOGICAL OPERATIONS AND DECISION MAKING, 59

Transition, *59*
New Resources, *59*
IF Statement, *60* Looping, *61* Sentinal Value, *63* Counters, *63*
Other Uses of Counters, *65* Accumulators, *66*
Combining Counters and Accumulators, *67*
Beware of . . . , *68*
Integrating the Old and the New, *69*
Exercises, *71*

6

LOOPING AND FORMATTED OUTPUT, 73

Transition, *73*
New Resources, *74*
FOR-NEXT Loop, *74*
Index Variables—Printing, Incrementing, and Stepping, *76*
FOR-NEXT with Branching, Counters, and Accumulators, *77*
PRINT USING Statement, *79* Tables, *81*
Beware of . . . , *83*
Integrating the Old and the New, *83*
Exercises, *85*

7

FUNCTIONS AND SUBROUTINES, 89

Transition, *89*
New Resources, *90*
LIBRARY Functions, *90* The ABS Function, *90* The INT Function, *91*
The RND Function, *92* The SGN Function, *93*
The Square Root Function, *94* User-Defined Functions, *94*
Subroutines, *96*
Beware of . . . , *98*
Integrating the Old and the New, *99*
Exercises, *100*

8

ARRAYS AND SUBSCRIPTS, 103

Transition, *103*
New Resources, *103*
Arrays, *103* Subscripts, *104* Subscript Manipulation, *108*
Counters and Accumulators, *108* Two-Dimensional Arrays, *110*
MAT Operations, *112*
Beware of . . . , *113*
Integrating the Old and the New, *114*
Exercises, *115*

9

STRING OPERATIONS, 119

Transition, *119*
New Resources, *119*
Character Strings, *119* String Variables, *120* String Assignment, *120*
LET, *120* INPUT, *121* READ, *121* String Output, *121*
Comparing Strings, *122* Subscripted String Variables, *123*
String Functions, *124* LEN, *124* Using String Functions, *127*
String Concatenation, *127* PRINT-USING Statements with Strings, *128*
Beware of . . . , *129*
Integrating the Old and the New, *130*
Exercises, *132*

10

SEARCHING AND SORTING, 135

Transition, *135*
New Resources, *135*
Searching, *135* Array Searching and Look-up Tables, *137*
Finding Maximums and Minimums, *139* Sorting, *140*
Sorting—Descending Order, *142*
Beware of . . . , *143*
Integrating the Old and the New, *144*
Exercises, *147*

11

DATA STORAGE AND FILES, 151

Transition, *151*
New Resources, *152*
Data Files, *152* File Processing, *152* Creating a File, *153*
The OPEN, PRINT, and CLOSE Statements, *153*
The INPUT # Statement and END Condition, *154* Searching a File, *155*
Updating a File, *156*
Beware of . . . , *158*
Integrating the Old and the New, *159*
A Final Note, *161*
Exercises, *162*

APPENDIX A

INTERACTING WITH THE COMPUTER, 165

Key Words vs. System Commands, *166*
The Apple II+ and II/e, *166*
The IBM PC and XT, *171*
TRS-80 Model III and 4, *177*
Timeshare Computer System, *182*

APPENDIX B

FLOWCHARTING AND PROBLEM SOLVING WITH BASIC, 187

Flowcharting, *187*
Alpha Electronics Repair—An Exercise, *194*

APPENDIX C

LANGUAGE REFERENCE, 199

Standard MINIMAL BASIC, *199*
BASIC PLUS, *200*
Dartmouth BASIC, *201*
Microsoft BASIC, *201*
Applesoft BASIC, *202*

INDEX, 205

1

Introduction to Computers

INTRODUCTION

Humans have long been fascinated with machinery. Brilliant scientists, engineers, and eccentric inventors have provided society with an endless array of machines methodically designed to relieve humans of tedious physical labor. The industrial revolution of the 1700s saw the introduction of machines that had been devised or invented to take advantage of the new supplies of power, namely steam and water. These machines enabled humans to amplify their own physical powers in ways never before possible. They could be used to shape, stamp, form, and machine raw materials into more useful items. As a result of the industrial revolution, the general quality of life for society increased and improved.

Now we are in the middle of a second industrial revolution. This time scientists, engineers, and inventors are providing us with tools that can be used to expand our mental capabilities rather than our physical ones. This new revolution is being called an information revolution. At the heart of it are computers that are capable of relieving humans of much of the tedious mental work involved in many of today's human endeavors.

History provides us with much information about colorful figures who attempted to invent machines that were capable of complex mathematical computation. The more significant inventions or machines include that ancient Chinese abacus, which is still used; the slide rule, which was invented by Edmund Gunter in the 1700s; the first mechanical adding machine, which was invented by Blaise Pascal in the 1800s; and the difference and analytical engines, which were conceived of by Charles Babbage in the early 1800s. These last machines were never actually built. The technology of those times was not capable of manufacturing the parts with a high enough degree of precision to enable the machines to work.

Actual data processing, the application of machines to the processing of

a large quantity of data at one time, was introduced in the late 1800s by Herman Hollerith, who invented the punch card to process data from the 1890 census. Punch card data processing remained the only method of processing large quantities of information until the early 1950s, when the first electronic computers were introduced.

Before the computer, most of the machines that were invented to aid in mathematical computation required tremendous involvement on the part of the operator. The electronic computer, commercially introduced in the early 1950s, was the first device that was capable of performing complex mathematical computations without the continuous involvement of an operator. In other words, computers then—and to a greater degree now—are capable of performing sophisticated, complex mathematical operations with a minimum of human intervention.

Today's electronic computers, while powerful computational aids, still require some degree of human involvement to direct their activities. Computers cannot perform the simplest of computations without a proper set of detailed instructions provided by a human being. These sets of instructions, called programs, are the subject of this book. Computers have a unique language that humans need to learn in order to be able to operate them efficiently.

In this book we explore one of the more popular programming languages called BASIC. Before involving ourselves in the technical details of programming, however, it is helpful to learn some background about computers and programming in order for the programming material to be of value to you.

The first two chapters of this book introduce general computer concepts and basic programming procedures, and several important techniques that you can apply to help you program more efficiently and accurately. After these two introductory chapters, we continue with the actual material for learning BASIC programming. If you have not already done so, read the Preface, which describes in detail the structure of the book and provides a framework for understanding how this book was written.

THE ELECTRONIC COMPUTER

The computer is by far the most powerful and sophisticated computing aid ever invented. While initially too expensive for everyone to have and use, with the advancements made in microelectronic technology, computing power has become available to anyone for as little as $100. Computers come in a variety of shapes and sizes. Not all of them are capable of the exact same activities, but even the smallest can provide considerable computing resources.

The best way to begin describing a computer is by comparing it to a biological computer—a human—and following the methods with which we perform data processing activities.

In order to activate a computer, we must first "Input" an adequate set of instructions. In our human systems we input the instructions either visually as written symbols or orally as spoken instructions. As humans, then, our eyes and ears act as input devices for our brain, which is our processing mechanism. Similarly, a variety of mechanical or electronic devices enter instructions into the central processing unit of the computer, which is named the *CPU*. The CPU performs the same functions for the computer that our

brains perform for us. It controls the entire system and performs all of the operations or activities.

Once our brains, or the CPU, have finished a process the results are usually communicated to the outside world in some understandable form. For us as humans this form can be written symbols or spoken sounds. For the computer this can take a number of different forms: a printed format for human use, an electronic form for use by the computer in the future, or even as speech output.

The next comparison with a human is more conceptual, as we may not be able physically to identify this component as easily. The computer has been designed to store and recall massive quantities of data. We can also store much data; however, all of our storage capacity or memory is located inside our processor, the brain. The computer has two separate storage facilities. The first, called *internal storage*, is located inside the CPU. It is used as a sort of "scratch pad" for the computer, doing all of its figuring or processing. This storage area is temporary and is used only while the computer is working on a particular task. Once a specific task is completed, this memory is erased to make way for a new task.

The second storage facility for the computer is called *auxiliary on-line storage*—auxiliary because it is not part of the CPU, and on-line because it is connected to it. Auxiliary storage acts as the computer's long-term warehouse, where instructions and data are stored until needed for some process.

Figure 1-1 is a schematic diagram of a computer system and the relationship of the components. Each block in this diagram represents one or more physical pieces of equipment that make up the computer system. Before moving on to the equipment discussion, we will use this diagram to describe in detail how a computer executes a given task.

FIGURE 1-1. The basic configuration of a typical computer system. The memory unit, control unit, and arithmetic logic unit are usually contained within a single cabinet. The functions of each of the components can be performed by a variety of machines.