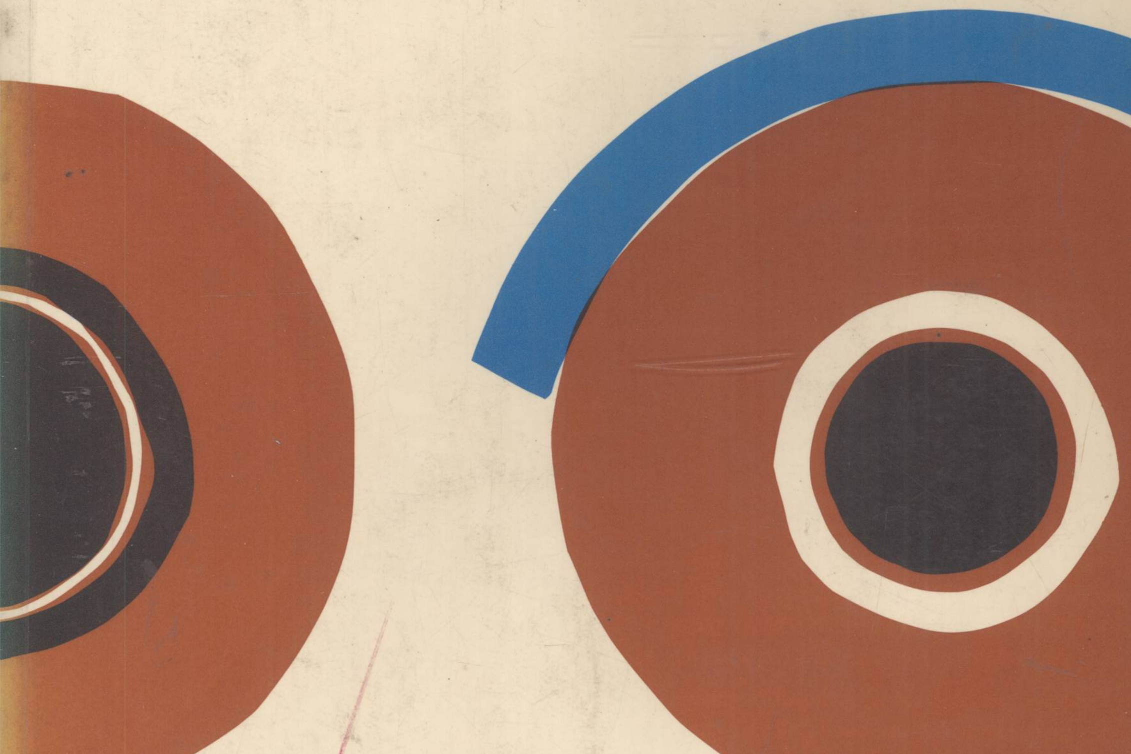
SPENCER

A Guide to BASIC Programming
2nd edition



A GUIDE TO BASIC PROGRAMMING

SECOND EDITION

DONALD D. SPENCER
Abacus Computer Corporation, Daytona Beach, Florida

ADDISON-WESLEY PUBLISHING COMPANY
Reading, Massachusetts · Menlo Park, California
London · Amsterdam · Don Mills, Ontario · Sydney

 Printed in the United States of America. Published simultaneously in Canada. Library of Congress Catalog Card No. 74-19707.

ISBN 0-201-07106-1
DEFGHIJ-AL-798

PREFACE TO THE SECOND EDITION

This book is designed for people who wish to learn how to use computers as problem-solving tools. It may be used by students in colleges and schools, and by practitioners in business and industry. It may also be used as a primary text in a computer problem-solving course, or as a supplementary text for courses in which the computer is used as a computational tool. Most of the book can be read and understood by readers who have had a year of high school algebra. Some of the problems, however, will challenge even the brightest reader. Whether the reader is using the book as a text in a formal course or as a self-teaching manual, the material included here provides the reader with all the knowledge and skills he will need for the development of effective computer programs.

More and more people today need to have a better understanding of the computer's influence on society. The computer has become so widely useful in business, schools, and industry that good opportunities can be missed without some training in computer technology. Until recently, computers have been either too expensive or too complicated to permit such training to be widely available to the nonprogrammer. Today, however, low-cost computing facilities are available in the form of minicomputers and time-sharing computer systems.

I believe that the BASIC programming language represents an excellent first (or only) language for most people to learn. It includes most of the computational components found in other programming languages, while requiring a minimum of attention to noncomputing details. BASIC is quickly learned. The average reader should be able to learn to write programs in BASIC with about five to ten hours of formal training or self-study and an equal amount of time spent studying the sample problems and working exercises.

Perhaps a novel feature of this book is the inclusion of a large number of problems from many diversified areas: mathematics, liberal arts, engineering, statistics, business, number theory, chemistry, accounting, physics education, game playing, and others. Problems were purposely chosen to illustrate that BASIC programming can be used to solve a variety of problems in a wide range of areas. Many problems are complete with flowcharts and BASIC programs. All programs in this book were checked out on minicomputer and time-sharing computer systems.

It is my hope that this book will create interest among students and practitioners and introduce to them some of the thrills, pleasures, and accomplishments that can be achieved with this marvelous tool we call a computer.

I am grateful to several computer manufacturers, equipment manufacturers, and universities for supplying me with information on specific BASIC implementations, for free use of computer systems, and for photographs of equipment and systems. I am also indebted to my wife, Rae, for her careful typing of the manuscript.

Ormond Beach, Florida D.D.S.
February 1975

PREFACE TO THE FIRST EDITION

This book is intended for the many people who would like to know what computer programming is about, especially programming on a time-sharing system. Computers have many important practical applications that should be of interest to other than computer programmers. There is much in the computer programming area that is of concern to engineers, scientists, students, and businessmen, as well as much that is informative and interesting to everyone.

More and more people today need to study computer techniques. The computer has become so widely useful in business, schools, and industry that good opportunities can be missed without some training in computer technology. Until recently, computers have been either too expensive or too complicated to permit such training to be widely available to the nonprogrammer. Today, however, inexpensive computers can be shared among many different people through a technique called *time-sharing*. In a time-sharing system, people communicate with the computer via a typewriter-like machine.

The majority of the present users of time-sharing systems fall into four catagories:

Students. In the field of education, time-sharing makes computers available to many students in universities, secondary schools, private schools, and industry. A time-sharing system provides an excellent tool for the student by permitting more hands-on experience and greater interaction with the computer. A student can learn a time-sharing programming language in hours instead of weeks and can prepare executable programs early in his training period.

Scientists, Engineers, and Mathematicians. The largest class of users consists of scientists, engineers, and mathematicians of industrial and research organizations, who are not professional programmers. They use the system as a computational tool because the languages are easy to learn and because they can prepare their programs near or in their working environment.

Businessmen. A third group of users of growing importance consists of lawyers, doctors, other nonprogramming professional people, and operators of small businesses, who are using time-sharing systems to help them in their everyday clerical tasks. Businessmen use *canned* application programs that have been developed by either computer manufacturers or time-sharing computer service firms.

Professional Programmers. Time-sharing systems are useful to professional programmers for debugging programs for later batch processing, for solving simple problems, or for running previously written application programs that are stored in the time-sharing library.

A computer must be programmed in some computer language. In order to simplify the programming task for nonprogrammers, *conversational programming languages* were developed. These languages resemble English and do not require the user to be burdened with knowing complex programming techniques. The simplest of all programming languages, presented in this book, is called BASIC, which is an acronym for Beginner's All-purpose Symbolic Instruction Code. The BASIC language is used primarily on time-sharing computer systems although it is also available for use on card-oriented systems. BASIC was developed under the direction of Professors John G. Kemeny and Thomas E. Kurtz at Dartmouth College.

BASIC is rapidly gaining acceptance as a general-purpose programming language. Though very simple, it is a very powerful language. Since BASIC is easy to learn and convenient to use, I am convinced that it will be included in the curricula of many schools and that it will also attract the interests of many engineers, scientists, businessmen, and laymen who may wish to study the language on their own. The average college freshman or high school senior should be able to learn to write simple programs in BASIC with about four hours of formal training and an equal amount of time spent studying the examples and working exercises. This book can be used equally well by the self-teaching individual, who can expect to understand the language in about ten hours spent reading the book and working several of the exercises. Whether the reader is using the book as a text in a formal course or as a self-teaching manual, the material included here provides the reader with all the knowledge and skills he will need for the development of effective computer programs.

Unlike most programming languages designed for professional computer programmers (PL/1, FORTRAN, ALGOL, COBOL, NELIAC, etc.), BASIC has only a few simple commands. A user sits at a typewriter-like unit and makes contact with a remotely located computer system as if he were the only user. (Actually, the computer is serving several other users in a similar manner.) A system such as this is called a time-sharing system. BASIC programming can be learned without access to a time-sharing terminal, but the reader is advised to use one if at all possible. It has been the author's experience that the computer is a never-tiring teacher that identifies areas no human teacher seems to find.

There are several versions of BASIC. Most are identical to the original Dartmouth BASIC, but some systems have added new language capabilities. This book is based on the General Electric BASIC language. Programs written in this version of the language can be run on GE 265, GE 420, and GE 600 Series time-sharing computer systems. They can also be run with little or no modification on the Burroughs B5500; Control Data 3300; Digital Equipment PDP-8/i and PDP-10; Hewlett-Packard 2000A; IBM System 360 (CALL/360: BASIC); RCA Spectra 70/46; and Scientific Data Systems 940, 945, and Sigma 7 time-sharing computer systems. Since only one version of BASIC is discussed, the distractions and confusion of trying to discuss several versions at once are avoided. The comparison of the various versions of BASIC given in Appendix A makes it possible to use this book in conjunction with any time-sharing computer system that has implemented the BASIC language.

Perhaps a novel feature of this book is the inclusion of a large number of programs from many diversified areas: mathematics, statistics, engineering, business, chemistry, physics, accounting, number theory, education, game playing, and others. Programs were purposely chosen to illustrate that BASIC programming can be used to solve a variety of moderate-sized problems. All programs in this book were checked out on the following computer systems: PDP-8/i, GE 265, GE 420, H-P 2000A, and SDS Sigma 7.

The first chapter of the text introduces time-sharing systems. Chapters 2 through 10 are devoted to BASIC programming. Chapter 11 includes 17 programs of varying difficulty from several different areas of interest. Regardless of whether the reader is a layman, high school student, college student, engineer, analyst, businessman, or professional programmer, he should find problems of particular interest to him. Chapter 12 includes 23 problems for reader solution.

At the end of the book are a list of references to other BASIC and related literature, and five appendixes. A minor feature that may be very useful to the beginning time-sharing user is an appendix that describes the operation of the Teletype Model 33 unit, the terminal device most widely used in time-sharing computer systems. Other appendixes include a BASIC comparison chart, a glossary of time-sharing terms, and a quiz of 50 review questions.

It is my hope that this book will create interest among students and nonprofessional programming personnel and introduce to them some of the thrills, pleasures, and accomplishments that can be achieved with this tool we call a computer.

I am indebted to several computer manufacturers, equipment manufacturers, time-sharing service organizations, and universities for information on specific BASIC implementations, for free use of their time-sharing computer systems, and for photographs of equipment and systems. Specifically, I would like to thank International Business Machines Corporation, Teletype Corporation, International Timesharing Corporation, General Electric Company, Scientific Data Systems, Com-Share, Burroughs Corporation, Digital Equipment Corporation, Hewlett-Packard, and the Massachusetts Institute of Technology. Finally, I am deeply grateful to my wife, Rae, for her careful typing of the manuscript, and to Miss Judy Munster for typing additions to and changes in the original manuscript.

Daytona Beach, Florida
October 1969

D. D. S.

CONTENTS

Chapter 1 INTRODUCING THE COMPUTER

Before a person can hope to understand programming, he must have some idea of what a computer is and how to use it. Thus we offer a brief description of the computer before we begin discussing programming and the BASIC language.

1.1 THE IMPORTANCE OF COMPUTERS

The computer is now becoming available to a large segment of the population and is beginning to have an impact on our everyday life. It is reshaping century-old ways of doing things. This machine, man's most remarkable invention, is invading every nook and cranny of society, opening up vast new possibilities by its extraordinary feats of rapid manipulation. It has made it possible to multiply by millions of times the capabilities of the human mind. In short, it is becoming so essential a tool, with so much potential for changing our lives and our world, that it is essential that everyone know something about it.

Computers have radically altered the world of business. They have affected military strategy, increased human productivity, made many products less expensive, and lowered barriers to knowledge. They have opened up new horizons to the fields of science and medicine, improved the efficiency of government, and changed the techniques of education.

Computers can store every variety of information recorded by man and almost instantly recall it for use. They can calculate tens of millions of times faster than the brain and solve in seconds many problems that would take batteries of experts years to complete. If the computer had not been developed, for example, American astronauts would have been unable to venture to the moon.

Computers have given science and technology the greatest tool ever developed for turning the forces of nature to human use. The reason is simple. The computer is more than a prodigy of information and analysis. It also never forgets what it has acquired. In time, it will even respond to oral command and report in both written and spoken English.

1.2 WHAT COMPUTERS CANNOT DO

Of all myths concerning computers, perhaps the most widespread and in the long run the most dangerous is this: *computers think for you.* A computer can no more think for you than can a screwdriver or the automobile you drive. Indeed, the reactions of a dog to external influences more closely resemble those of a man than those of a computer do; but who would expect his dog to *think* for him?

Although the computer is extremely fast in its calculations, it is absolutely incapable of doing anything more than carry out the instructions that have been written for it and placed into it by human beings. It has no more intelligence than a hammer or a rock and is totally useless without human direction.

Can you conceive of a computer inventing its own problems? No; computers do not think and therefore they cannot create their own problems. They can never reproduce or replace the human thinking process. Computers can manipulate and combine items of *data,* but they cannot, beyond the strict limits of their instructions (programs), infer any meaning or *information* from this result. Human beings do this with great facility, both consciously and unconsciously.

A computer must be instructed (by a computer program) to perform a series of logic and arithmetic operations, by a human being who knows the *language* of the computer. The human being, then, is the agency that commands the computer. The computer is his electronic slave, just as the huge elephant working in a tropical forest is the slave of his mahout. The man does the thinking; the computer or the elephant does the work.

The computer is a willing slave and will perform whatever it is instructed to do. However, it will produce a million incorrect answers just as fast and as willingly as it will compute a million correct ones. Human beings must thus exercise great care in preparing instructions for the computer. These must be absolutely unambiguous, precise, and complete to the last detail. If they are not, the computer is very likely to produce "unexpected" answers. Unexpected answers are often mislabeled "creative answers" because they provide information that was previously unknown to the user and thus give the appearance of original thought. All the computer has actually done, however, is to carry out the instructions that were specified beforehand. The reason that the results are a surprise is probably that there were so many calculations to be made or facts to be analyzed that no one could estimate what the outcome would be.

By now the reader should be convinced that certain human functions are indispensable to the operations of computers. It is a human being who must conceive the need for the computer's work, who must prepare the instructions for that work, and who must put the whole process into motion. It is supremely unlikely that computers will ever pose an unmanageable challenge to man. For all his shortcomings, man is uniquely capable of responding to unforeseeable contingencies for which there is neither precedent nor experience. Instinct, intuition, and inspiration have raised man to his highest peaks, and mere computer logic will never attain comparable levels.

1.3 WHAT IS A COMPUTER?

When someone asks, "What is a computer?" we have to reply, "What kind of computer do you mean?" The reason for this is that there are two basic types of computers: the *analog* and the *digital*.

Analog computers solve problems by measuring such continuously occurring variable physical conditions as pressure, voltage, temperature, speed, or position and translating these measurements into a continuous, meaningful output form. Several simple examples of analog devices are the slide rule, automobile speedometer, thermostat, hourglass, and thermometer.

The *digital computer*, the most versatile machine of the electronic computer family, performs calculations by counting (by its nature, a discrete, *noncontinuous* operation). The digital computer can represent information only as discrete numbers and works on data mathematically rather than physically. There are two types: the *special-purpose computer*, which is designed for only one application, and the *general-purpose computer*, which may be used in many different applications. (Most general-purpose computers are of the *stored-program* type, that is, they store their instructions in their internal memory.)

1.4 DIGITAL COMPUTER ORGANIZATION

Although digital computers come in many shapes and sizes, they are all similar in many ways. Each computer must be able to *read in* instructions and data, *remember* the problem being solved and the data to use, *perform* calculations (and other manipulations) on the data, *read out* the results, and *control* the entire operation. Thus, for a machine to process data it must contain five logical elements:

1. A means of input
2. A means of output
3. An arithmetic unit
4. A means of storing data
5. A control unit.

These elements all work together in solving a problem; numerical data and instructions are constantly being sent back and forth between them. This entire process is under the control of instructions that are specified in a *program*.

Input to the computer may consist of any type of data: commercial, scientific, engineering, statistical, and so on.

The *central processing unit*, sometimes called the *central processor* or CPU, is the heart of the computer system. It consists of three logical parts: the arithmetic unit, the storage unit, and the control unit. The central processing unit is often called, simply, the *computer*.

The *arithmetic unit* performs all the arithmetic required by the computer program, that is, the four basic operations of addition, subtraction, multiplication, and division. These operations are in turn used in a variety of ways to perform other calculations. The arithmetic unit can also perform logical operations such as comparing two items of data for alphabetical or other sequence.

The *storage unit* stores the program and data while the computer is working with them. The program and data are inserted into the storage area, often called the *memory*, through the input unit. While the computer is solving a problem, the program instructions and data might continually pass back and forth between the arithmetic unit, the control unit, and the storage unit. After the program is finished, the processed data is transferred from the computer through the output unit.

The *control unit* performs the most vital function in the computer. It directs the overall functioning of the other units and controls the data flow between them during the process of solving a problem. The control unit in turn is controlled by the program; in other words, it merely performs the instructions as specified in the program. Thus a *computer program* is a set of instructions telling the computer what to do. These instructions must be assembled in a logical manner in order to do a specific job.

An *input device* is a machine that can accept direct input from a keyboard (such as a typewriter keyboard) or that can read previously recorded data from punched cards, magnetic-ink character printing, punched paper tape, magnetic tape, printed documents, voice recognition devices, and other forms. These input devices are an integral part of a computer system and operate under the control of the central processing unit as directed by the computer program. As data is read by the input device's sensing mechanism, it is converted through electronic pulses to magnetic recorded form.

An *output device* reports information from computer storage in a form that can be understood by human beings or that is suitable for use as input in another computer system. Commonly used output devices are typewriters, printers, card punches, display devices, paper tape punches, magnetic tape units, magnetic disk units, and audio response units. Instructions in the computer program may select a specific output device.

1.5 HOW THE COMPUTER WORKS

The computer is a piece of machinery to be used as a tool. However, most other tools are easier to use; it is usually possible to install them, read the instructions, start them, and get promptly to work. It is not possible simply to push a computer's start switch, however, and have it perform useful work. A newly installed computer might be compared at best to a five-year-old child, for a child is at first capable of doing very little, but he can by degrees be taught to do a variety of useful things. The more a child is taught, the more complex the work he can do. The computer can also be taught to do many things, but only by a process involving considerable planning, effort, and time. The "teaching" of a computer must be done through the form of programs. Because it can respond only to a basic set of simple instructions, problems for the computer must be broken down into a great number of detailed steps.

The following analogy will help the reader understand the extent to which problems for computer solution must be broken down. Suppose that you wanted to add two numbers, say, 36 and 41. This operation certainly sounds simple enough. However, let us examine the thought process required to perform it. First you must determine that you are going to add two numbers. Do you know how to add? If not, you must learn before proceeding. Next you must obtain the numbers to be added. If they are unavailable, then you cannot add. Write down the first number. Write down the second number. Are the numbers aligned correctly? If not, then erase the second number and rewrite it. Repeat this process until the numbers are aligned properly. Then draw a line below the numbers. You are now ready to add the numbers to compute the sum.

This set of instructions for a simple written addition is roughly analogous to the program of instructions that must be provided for a computer. Every process, no matter how elementary, must be broken down into simple steps that the computer understands.

A simplified program might, for example, be the finding of the value of $y = x^2 + 6$ if x is equal to 4. The program would contain the following steps:

1. Place the value associated with x and the constant 6 into computer storage.
2. Bring the value of x from the storage unit into the arithmetic unit.
3. Multiply the value of x by itself, thus forming x^2.
4. Add the number 6 to the product obtained in step 3.
5. Print the answer obtained in step 4.
6. Stop.

This program, written in a language that the computer understands, would originally be read into computer storage via the input unit, step by step, in the same order. After all the program was in the storage unit, the computer would start processing the program in the following order: step 1, then step 2, and so forth, until the sixth step was reached. At this time the problem would have been solved and the result printed.

1.6 EQUIPMENT IN A COMPUTER SYSTEM

A *computer system* consists of a number of individual components, each of which has its own function. The system consists of units to send information to the computer (such as the typewriter shown in Fig. 1.1), the computer itself, auxiliary storage units, and equipment to accept information from the computer (such as the keyboard display device shown in Fig. 1.2).