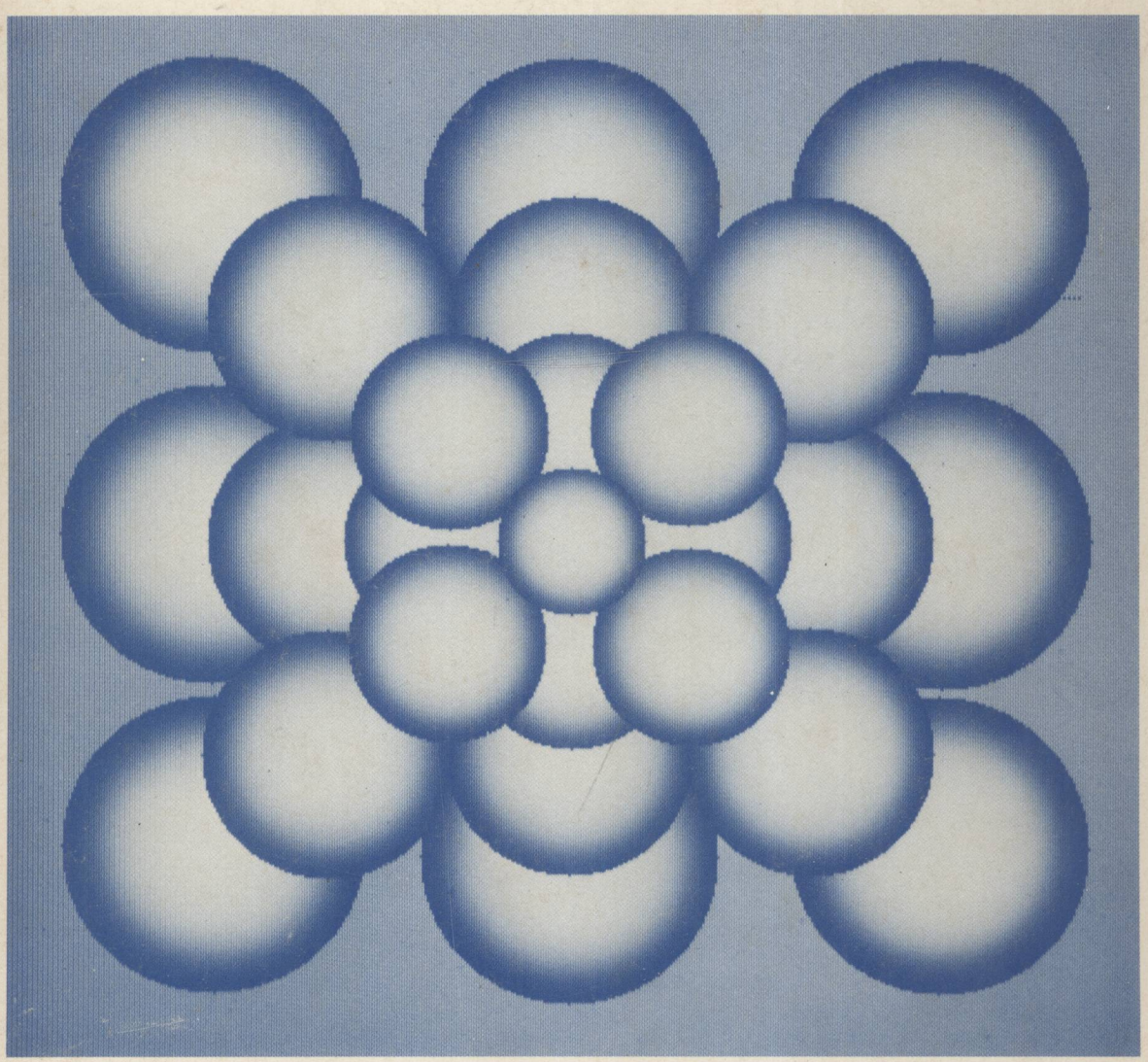
A Structured Approach to BASIC Programming



C. Joseph Sass

A Structured Approach to BASIC Programming

C. Joseph Sass
University of Toledo

Allyn and Bacon, Inc.
BOSTON · LONDON · SYDNEY · TORONTO

Production Editor: Jay Alexander
Art Editor: Armen Kojoyian
Mfg. Buyer: Linda Card

Portions of this book first appeared in *BASIC Programming and Applications* by C. Joseph Sass, copyright © 1976, by Allyn and Bacon, Inc. and in *BASIC Programming for Business* by C. Joseph Sass, copyright © 1972, by Allyn and Bacon, Inc.

Library of Congress Cataloging in Publication Data

Sass, C Joseph.
A structured approach to BASIC programming.

Includes bibliographical references and index.
1. Basic (Computer program language) I. Title.
QA76.73.B3S29 001.6'42 79-17040

ISBN 0-205-06726-3

Printed in the United States of America

Printing number and year (last digits):
10 9 8 7 6 5 4 3 2 85 84 83 82 81 80

Preface

The decreasing cost of computing equipment has led to a dramatic increase in the number of computer installations in small and medium sized companies. It has also led to a corresponding increase in the number of business persons, educators, engineers, social, medical, and government workers, researchers, and scientists who are directly involved with computer systems and data processing services in general. This text is directed to those interested users and potential users of computer facilities who desire to learn a programming language to further their knowledge of computer systems, equipment, and capabilities. This knowledge is obtained and expanded upon by thoroughly covering the programming language BASIC ® (*B*eginner's *A*ll-purpose *S*ymbolic *I*nstruction *C*ode).

While the primary intent of this text is to discuss the BASIC programming language in detail, other significant, related data processing concepts are also presented. These topics, including flowcharting fundamentals, logic development, structured programming, pseudocoding, file organization and manipulation, and systems flow, are integrated with the language and are discussed clearly and concisely.

Since BASIC, plus the extensions to the language that are presently available in many versions, is a powerful, yet easy to learn language, this book can be utilized in a variety of ways: (1) It can serve as the primary text in a formal graduate or undergraduate course with instructional aid, (2) as a self-teaching manual for readers who are learning the BASIC language on their own, (3) as a reference source for current users of the language or, (4) finally, as a supplementary text for an introductory data processing course that covers the BASIC language.

This version of the text incorporates three pedagogically sound concepts that have proven helpful to readers, especially novice programmers. First, complete computer program samples are illustrated throughout each of the chapters. These samples are shown with the actual computer listing of the program and the generated output. Each of the programs illustrates a particular use, function, or programming concept. The programs are formulated under structured control, using only simple, selection/decision, or iteration/loop logic patterns. One-half of the programs are accompanied by a flowchart that describes the general flow of logic followed in the solution; while the other half of the program samples are accompanied by pseudocode that takes the place of a flowchart. The samples envelop a wide variety of applications, including business, education, engineering, mathematics, and social science. The program solutions offered for discussion purposes in the early chapters are straightforward. As new, more complicated material is introduced, the solutions are expanded and become more involved, thereby providing the reader with the opportunity to trace the logic flow upward from simple structures to large, complex applications in the later chapters.

The second concept relates to a series of figures that are an integral part of the discussion of new material in each chapter. These figures describe and list the common errors that the novice programmer is likely to encounter; and, finally, present a corrected version of the erroneous statements that have been illustrated.

The third idea pertains to the organization of the text. Each chapter is subdivided into logical sections, with each section being concerned with only one major topic. While the chapters are progressive in terms of complexity, several sections are optional and may be omitted without loss of continuity. A set of exercises follows each of the sections and relates primarily to the main topic covered. Answers to many of the exercises are listed at the back of the book, thereby permitting the readers to check their comprehension of the section material. More difficult questions are found at the end of each chapter, with selected solutions again available for reader analysis at the back of the book.

Many versions of the BASIC language are available from commercial vendors and manufacturers of time-sharing and small computer systems. In this context the author has endeavored to present only statements that are fundamental to most dialects of the language and the American National Standards Institute version. An exception to this mission relates to statements that pertain directly to the structured programming and pseudocode concepts. And when the exceptions do occur, the material is either optional or a number of footnotes highlight and clarify the differences in use and structure from system to system. Thus, the book is neither machine nor vendor oriented; rather, it is adaptable to many different systems.

A brief introduction to computer systems, time-sharing, programming, flowcharting, structured programming, and pseudocode is covered in Chapter 1. Chapter 2 deals with the elements of the BASIC language and simple structured programs. Chapter 3 introduces several input methods, loop/iteration structures, and internal program documentation. Logic branches and selection structures are included in Chapters 4 and 5. The optional IF-THEN-ELSE statement, which is particularly important to structured programs, is also included in Chapter 5, along with more advanced printing features. Loop con-

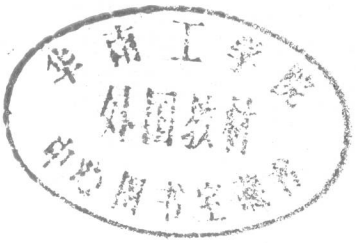
trol is covered in Chapter 6 with the standard FOR-NEXT instructions and the optional FOR-UNTIL and FOR-WHILE statements that are utilized in structured programs. Subscripted variables are introduced in Chapter 7. Also included are optional sorting and advanced structured examples. Matrix commands are discussed in Chapter 8; and internal, plus user-defined functions are in Chapter 9, followed by subroutine structures in Chapter 10. The use of files for data storage, recall, and manipulation is illustrated in Chapter 11; while important supplementary information is provided in the appendices.

The material in the first seven chapters is contiguous in nature and, normally, will be covered in a one quarter undergraduate course. Chapters 8 through 11 can be introduced in a different sequence or included in the material to be discussed in a semester or in a graduate course. Chapter 9, in particular, may be introduced in an earlier sequence without a loss of continuity.

The author is indebted to a number of people and corporations who have helped make this book possible. Time-sharing and small computer systems manuals, photographs, and services provided by the following companies and institutions are greatly appreciated: American Telephone and Telegraph, ACTS Computing Corporation, Burroughs Corporation, Control Data Corporation (Service Bureau Division), Digital Equipment Corporation, General Electric Corporation, Hazeltine Corporation, Hewlett-Packard Company, Honeywell Information Systems, Inc., International Business Machines, Inc., J. Preston Levis Regional Computing Center, The University of Toledo, Teletype Corporation, Texas Instruments, and Wang Laboratories. Special thanks go to the reviewers of the original manuscript, to past students who have made many helpful suggestions, to Gail Short and Rick Bonitati who aided considerably with the solutions manual, and to a very patient wife who endures both the writing and re-writing.

C.J.S.

Contents



Introduction to Time-Sharing and BASIC Programming

1

The dynamic growth of computer applications and the corresponding decrease in computer costs have changed the operations of many diverse groups in our society. Every day new and different problems are being solved with the aid of a computer system. One of the reasons for this is that computer systems costing $100,000 five years ago now cost substantially less. In fact, national retail stores now sell entire computer systems for less than $1000. These are completely functional machines that can be productive tools. In addition to the traditional uses of computers in business, engineering, government, and research, computers are now employed in dissimilar fields such as education, crime prevention, traffic control, ecological studies, and even bookkeeping tasks at home.

Whenever data or information is gathered for the purpose of study and further investigation, a computer* is beneficial. It has the processing capability of compiling, storing, correlating, and selecting data rapidly and accurately. Through the storage of instructions and information, simple or complex formulas and algorithms can be employed to transform voluminous data into concise, meaningful statistics.

This chapter introduces the reader to the fundamental concepts of computer systems, hardware, programming, and flowcharting.

1–1 Functions of a Computer System

The computer processing cycle common to all systems is comprised of three operations: the *input* cycle, the compute or *processing* cycle, and, lastly, the *output* cycle. This series of operations is performed by a computer system. Five

*See Appendix A for a brief history of computer development.

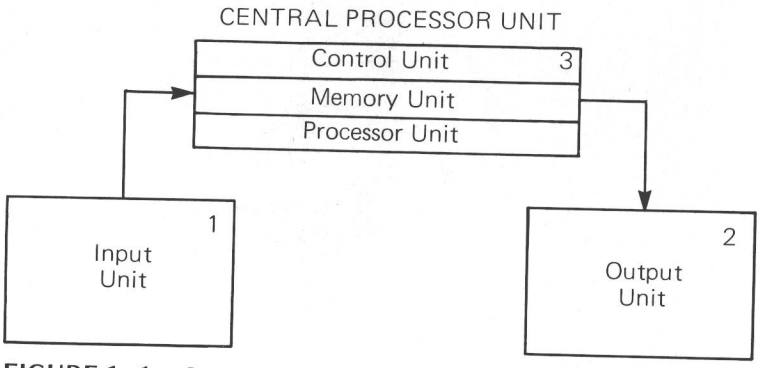


FIGURE 1–1. *Components of a Typical Computer System.*

important physical components comprise any computer system (Fig. 1–1); often these are classified singularly as computer *hardware.* To begin processing, information or data must be transmitted to the computer to permit mathematical operations or logical manipulations of the data to be performed within the computer. The INPUT unit satisfies this need. Devices used for transmitting data or input into the computer include keyboards, card readers, paper tape readers, and several types of magnetic tape readers, such as cassettes. In general, the hardware that function in communication with the computer are identified as *peripheral devices.*

Whereas the INPUT device passes information to the computer system, the OUTPUT unit serves to transmit data, normally answers, from the computer to the user. Examples of output peripheral devices are card punches, line printers, terminals, and several forms of magnetic tape and disk.

The final three components of a computer system, the MEMORY unit, the CONTROL unit, and the PROCESSOR unit, are also called the CENTRAL PROCESSOR UNIT (CPU). Often, the three are physically housed in the same unit, though they perform individual operations. The MEMORY unit retains or stores for later use the data or information that is transmitted to the computer by the INPUT device. It also stores the results of internal computations. In other words, the MEMORY unit serves as a retention device for data entered into or computed by the system. The CONTROL unit performs the necessary function of coordination. The activities and operations of the entire system are monitored and directed by the CONTROL unit. Thus, it assures that processing steps are executed as dictated by the user. The last component is the PROCESSOR unit or, as it is sometimes identified, the arithmetic unit. The mathematical and logical steps performed by the computer are carried out in the PROCESSOR unit. For example, the addition, subtraction, and the logical or decision tests of a set of instructions are made within the PROCESSOR unit.

Together, these five components operate as a computer system (Figs. 1–2 and 1–3). The five units, when combined with a computer program, can be used to advantage in solving a variety of different problems. A computer system should be regarded on this basis—as a specialized tool that can provide many answers for management and other users of the system.

1–2 The Start of Time-Sharing

One segment of the computer field is providing computer access to many remote users by a concept called time-sharing. This is the utilization of one or

FIGURE 1–2. *Large-Scale Computer System. (Courtesy of International Business Machines, Corp.)*

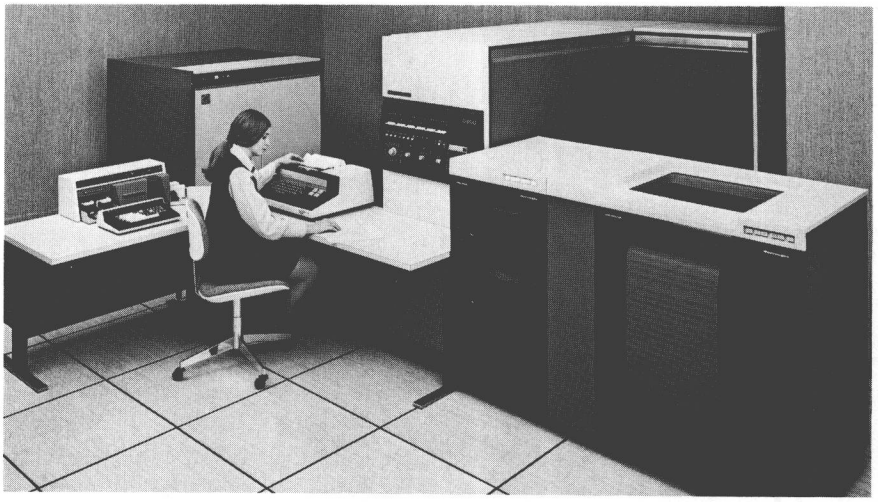


FIGURE 1–3. *Small Computer System. (Courtesy of the Burroughs Corp.)*

more central computer systems by many different users and customers, each located a distance from the computer and having a fixed amount of computer time allocated on both a rotating and priority basis. Thus, many users can effectively share one central computer system almost simultaneously. The customer normally has access to the computer by means of a typewriter-like device called a remote *terminal,* connected via telephone lines to the central

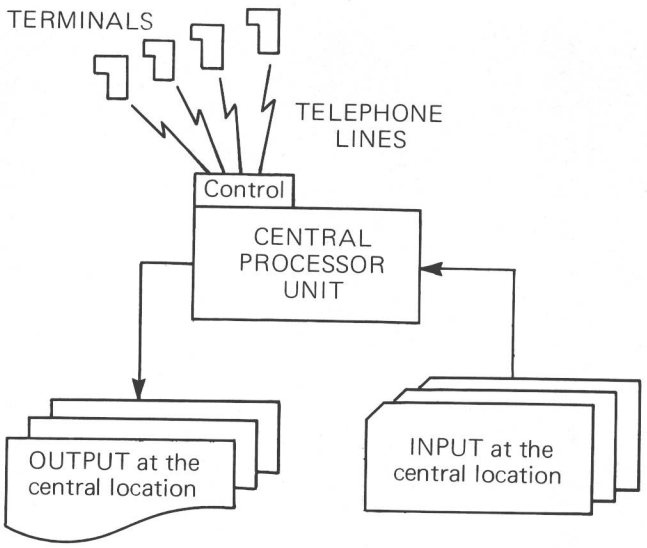


FIGURE 1–4. *Hardware Requirements of a Commercial Time-Sharing System.*

computer (Fig. 1–4). The time-sharing user is linked to the computer by the remote terminal as an INPUT/OUTPUT device; and since the connection is usually via telephone lines, this implies that the main computer system may be found at other physical locations. Frequently, the computer system is, in fact, located at another site, city, or even state.

Why use time-sharing? One advantage is cost. Since many customers have access to the same computer system, the cost likewise is shared. A second advantage is the speed with which a response from the computer is obtained by the user, especially if other users have arrived beforehand. With other types of computer systems, a longer response period may be involved. A third advantage is the ease of operation. One popular terminal utilized in the time-sharing mode is the Model 33 ASR Teletype (Fig. 1–5), although many different manufacturers supply terminals for this purpose.* To learn how to operate these terminals is relatively simple. A fourth advantage of time-sharing is the conversational mode of INPUT/OUTPUT permitted. Conversational mode is the ability of the programmer or the person operating the terminal to actively communicate in a question and answer response environment with the computer.

The fifth advantage is that the programming language, BASIC, available with time-sharing, is one of the easiest programming languages to learn and use. Many of the key statements have an inherent English connotation. Therefore, the language can be understood with minimum effort in comparison to other languages.

How did time-sharing and the BASIC language originate? One of the first time-sharing systems was developed at the Massachusetts Institute of Technology. A year later, in 1962, the company of Bolt, Beranek, and Newman furthered the advancement of time-sharing by developing a system for the Digital Equipment Corporation. In 1963, the Rand Corporation in California announced a time-sharing system that provided the programming language, JOSS (*J*ohnniac *O*pen *S*hop *S*ystem). Later, the JOSS language became part of

*General Electric, Hazeltine, and Texas Instruments are just three of the many companies that supply terminals (see Figs. 1-6, 1-7, and 1-8).

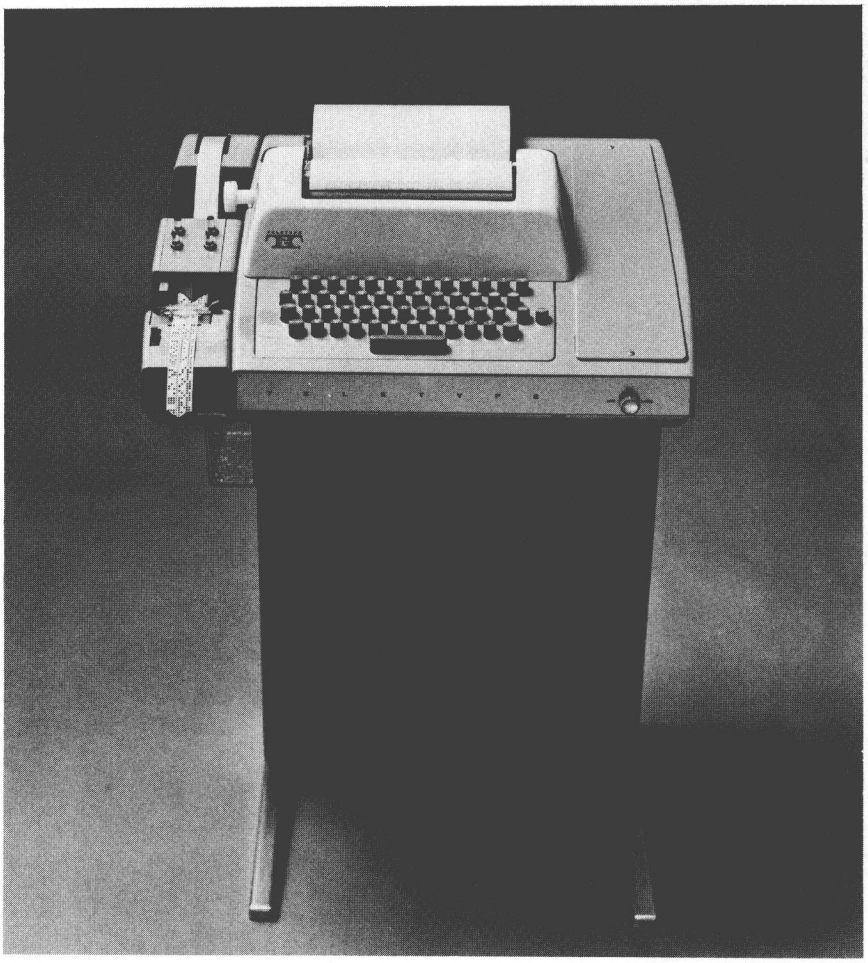


FIGURE 1–5. *Teletype Model 33 ASR Data Terminal. (Courtesy of Teletype Corp.)*

the foundation for BASIC (*B*eginner's *A*ll-purpose *S*ymbolic *I*nstruction *C*ode). BASIC was developed on a time-sharing system at Dartmouth College during the period from 1964 to 1966. Professors John G. Kemeny (now President of Dartmouth College) and Thomas E. Kurtz created the language and were the authors of the first book on BASIC.

How does BASIC differ from other programming languages? Many times the time-sharing user does not want to become a programmer. What the user does want is the capability to utilize the computer as a tool in fulfilling the objectives for a particular job at hand. Therefore, the user is interested in learning a programming language that permits the maximum benefit to be derived from the computer but in a manner not technically beyond understanding. BASIC is a language that serves these purposes. Yet, sophisticated applications have been programmed in BASIC.

Whereas BASIC started out as a language employed predominantly on time-sharing computers, with the advent of smaller, less expensive systems BASIC has grown in popularity to the point where it is the most commonly used language for systems that cost $25,000 or less. These small computers, or minicomputers as they are frequently called, are used in businesses, schools, and even homes. It is likely that this trend toward BASIC will continue as long

FIGURE 1–6. *TermiNet 300 MSR Split Platen Printer. (Courtesy of General Electric Company.)*

FIGURE 1–7. *Hazeltine Cassette Recorder, Cathode Ray Tube, and Printer. (Courtesy of Hazeltine Corp.)*

as the costs of processing continue to decrease. One of the major differences between time-sharing and small computer processing is that in the former the resources of a larger system are normally shared with others, while in the latter the small computer normally only supports one operator at a time in what is

FIGURE 1–8. *New "Silent 700"® Model 735 Portable Terminal from TI. (Courtesy of Texas Instruments, Inc.)*

called a *batch* environment. In a simple batch environment one job is processed at a time, followed in sequence by a group or batch of jobs that are processed sequentially in the order that they appear.

1–3 Accessing the System

How does one become a time-sharing or small computer user, programmer, customer, or operator? Two requirements must be fulfilled prior to becoming a user: (1) A contract must be concluded with a time-sharing vendor or another source of service made available (i.e., an in-house computer system); and (2) a suitable terminal or keyboard device must be installed. In this text, discussion of terminal operation will be limited primarily to the Model 33 ASR Teletype.

Having completed the proper set-up for initial processing, three final steps are necessary to access and use the system:*

1. Turn the terminal power unit on and dial the telephone number of the time-sharing computer system. Or, with other installations, simply turn the power ON to activate a direct connection to the computer. Once on-line, the computer will respond in a conversational mode by typing information on the terminal and requesting specific answers in return from the operator. When this sign-on procedure has been completed, the computer will indicate that it is READY for the operator to continue.
2. Enter your program and/or system commands into the computer via the terminal keyboard.

®Trademark of Texas Instruments, Incorporated.

*Detailed information for a typical time-sharing sign-on procedure is provided in Appendix B, Terminal Operations.