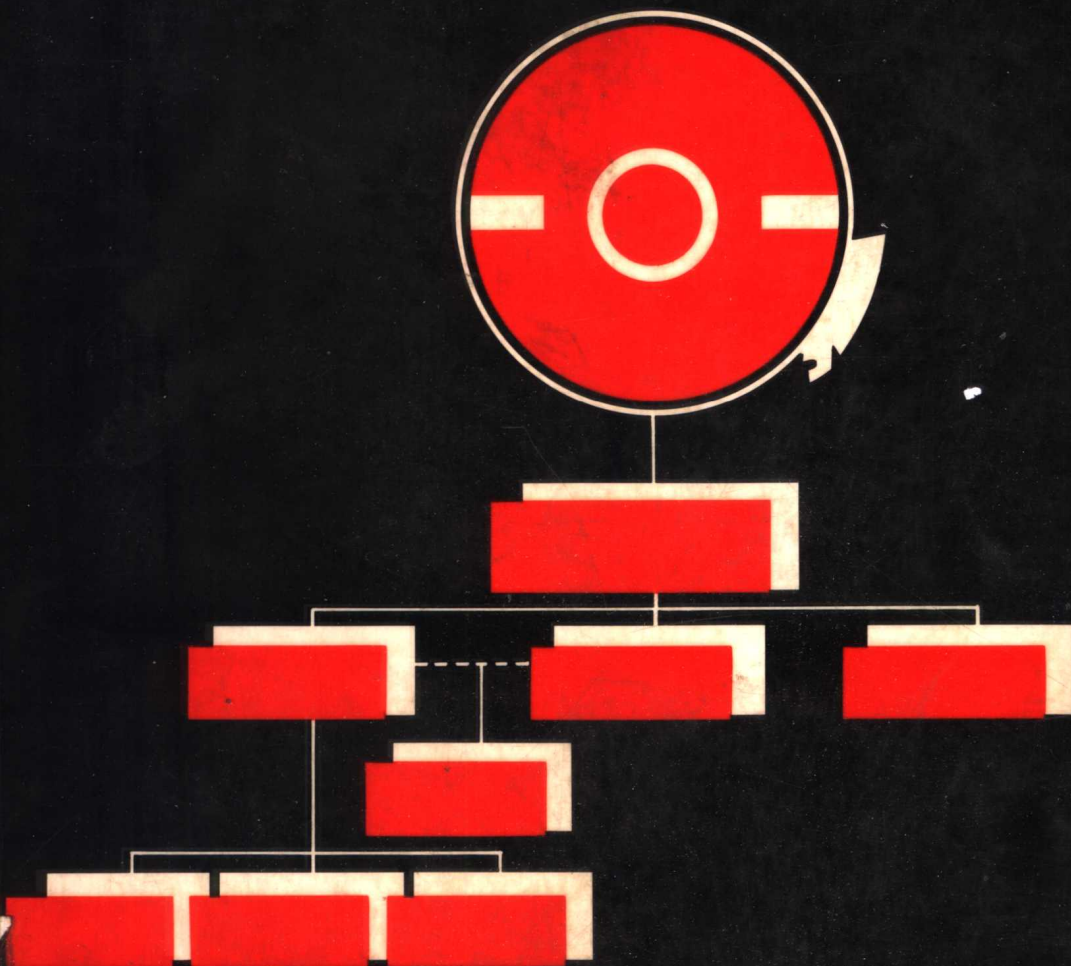


# STRUCTURED VAX BASIC AND BASIC-PLUS

2ND EDITION



TEGLOVIC AND DOUGLAS

***STRUCTURED VAX  
BASIC  
AND BASIC-PLUS***

**Steve Teglovic, Jr.  
University of Northern Colorado  
Kenneth D. Douglas  
Southwest Missouri State University**

**Second Edition 1987**

***IRWIN***

**Homewood, Illinois 60430**

---

# *Preface*

---

With the increasing use of minicomputers and personal computers, more individuals and organizations are acquiring computer systems where BASIC is the predominant language. In fact, industry reports show that BASIC is the most used language for mini- and microcomputers today and is one of the more popular languages for medium- and large-scale computers. BASIC has had strong support from both hardware and software vendors.

## **Text Objectives**

---

Because of BASIC's high use, this text was written to meet four major objectives for the potential BASIC programmer:

1. The material is presented using a modular approach, starting with the very basic concepts and skills needed, and progressing into intermediate and then advanced concepts. The programmer can quickly learn to write meaningful programs, to develop programming techniques, and to gain confidence in the use of the language.
2. Although BASIC was not designed as a structured language, the authors have used structured programming techniques as an approach to program development. This should result in better program design and programs that are easier to test and maintain. Structured programming is the state of the art approach used in most data processing applications today.
3. Since BASIC is not a standardized language and many different versions of it exist in the world of computers, the authors have decided against using a generic BASIC that does not satisfy any one computer's need and have used a machine specific version for DEC computers. The advantage of this approach is that every example, program, and exercise has actually been run on a computer and the reader can study the results to gain a better knowledge of how the language really works.

4. With but a few changes in the version of BASIC used in this text (such as the PRINT USING and IF/THEN/ELSE statements), the reader can use this text for programming any computer that has a BASIC language. For example, the BASIC used on IBM personal computers is quite compatible with the BASIC in this text.

## Approach to the Text

---

The text is designed for individuals who are just learning to program, or for those already familiar with a programming language. In addition to using a modular approach and structured programming techniques, there is extensive use of structure charts, flowcharts, pseudocode, and BASIC coding. Most of the chapters contain fully developed programs to illustrate the material covered in each chapter.

All programming statements are fully illustrated and explained. They are used by sample statements and example programs. The examples show applications in administration, business, science, and the social sciences. Near the end of most chapters there is a section on common errors that occur when using the statements covered.

## Text Examples, Programs, and Exercises

---

There are about 100 examples in this text that have been written and tested on a computer. In addition there are over 35 complete sample programs that include: problem description, input, output, structure charts, a flowchart and/or pseudocode, BASIC code, and program execution. There are also 165 exercises. The exercises are included at the end of chapters and range in difficulty from very simple to hard. An instructor's manual is available with the text. It has actual programmed solutions for all of these exercises. The manual will save many hours of programming and preparation time.

## Changes from the First Edition

---

Whereas in the first edition of this text the primary emphasis was on BASIC-PLUS and secondary emphasis on VAX BASIC, the reverse is true in the revised edition. This is because many organizations upgrade to a VAX system from a PDP system.

The old Chapter 11 on subroutines has been merged with Chapter 9 and the chapter on text editing has been placed in Appendix E. The material on structured programming has been greatly expanded and placed earlier in the text in Chapter 4. The use of structure charts for planning and documentation begins in Chapter 5. Table processing has

been divided into two separate sections: one to cover one-dimensional concepts and the other for two-dimensional concepts.

The use of a second color in the revised text highlights important concepts, ideas, and techniques. It also enhances references to certain text material and text readability.

## Organization of the Text

---

The text uses a modular approach to the study of the BASIC language. Simple concepts are covered first. As new statements in the language are needed, that material is brought in. At the end of each chapter, the previous material in the language is integrated along with the chapter's new material in program illustrations.

Section 1 is an introduction to using the BASIC language and a computer system. Emphasis is placed on using an operating system and executing programs that have already been written. The purpose is to get the reader acquainted with computer operations before it is necessary to understand the language's syntax and the logic of programming. In addition, the basic components of the language are introduced. Structured programming concepts as well as the use of structure charts and pseudo-code are discussed.

The thrust of Section 2 is to learn the basic logic of how to enter, process, and output data. Use is made of simple forms of the INPUT, READ, DATA, LET, and PRINT statements. Some decision making is covered with simple IF statements.

Section 3 builds on the concepts from previous sections and emphasizes the use of structured programming. More difficult programming techniques are described for assignment, output, and decision statements. New instructions for subroutines, intermediate looping, table processing, and statement modifiers are discussed.

Section 4 covers the more advanced topics of function subprograms, character manipulation, and matrix operations. File processing is also discussed. More specifically, the techniques of sequential processing, random processing, and block input/output processing are covered in some detail.

Appendixes at the end of the text cover in detail: BASIC reserved words, ASCII character codes, VAX BASIC and BASIC-PLUS error messages, recoverable error processing, and text editing.

## How to Use the Text

---

The text can be used in a variety of ways. Chapters 1 through 9 or 1 through 12 can be used as a supplement for an introductory text to information processing or computer literacy. *The entire text can be used as a*

stand-alone text for teaching a standard BASIC programming course. Chapters 4 through 15 can be used to teach more advanced BASIC concepts for those that have already been exposed to simple BASIC programming.

## Acknowledgments

---

The authors' special thanks go to the many, many users of the first edition of this text. Their comments, suggestions, and evaluations made about the text have greatly influenced the outcome of the revised edition. Our thanks also go to the many reviewers of both the first and revised edition of the text for their help on the technical matters during the writing and production of the text. Specifically, we would like to thank the following reviewers of the second edition: John K. Cartee, Jackson State Community College, Jackson, Tennessee; William Carlborg, Prairie State College, Chicago Heights, Illinois; David C. Miller, University of North Carolina at Asheville, Asheville, North Carolina; and Anne L. Kerfoot, Shepherd College, Shepherdstown, West Virginia.

Textbooks are seldom written without the support, patience, and timely words of encouragement from others. Therefore, we would like to express our greatest and most sincere appreciation to our wives, Mary and Cecil, two of the greatest nurses we know.

**Steve Teglovic, Jr.  
Kenneth D. Douglas**

---

# *Contents*

---

## **SECTION I**

<b><i>Introduction to the BASIC Programming Language</i></b>	<b>1</b>
<b><i>1 Introduction to Computers and the BASIC Language</i></b>	<b>2</b>
Functions of Computers	2
Processing Tasks	3
Central Processing Unit (CPU)	3
Input, Output, and Storage Devices	5
Terminals	5
Line Printers	5
Magnetic Tape	5
Magnetic Disk	5
Batch versus Interactive Processing	6
The BASiC Language	6
Levels of Languages	7
How BASIC Works	7
Program Development	11
Exercises	11
<b><i>2 System Commands</i></b>	<b>13</b>
Logging on and off the System	13
VAX Procedures	13
PDP Procedures	14
Entering a Program	17
Listing a Program	19
Running a Program	20
Storing a Program	20
Accessing a Stored Program	21

Deleting a Stored Program	22
Storing a Modified Program	22
Getting a Directory or Catalog	26
A Typical Session on the Terminal	26
Other System Commands	26
Making Corrections and Modifications	26
Exercises	33

### **3 Components of BASIC**

**36**

Constants	36
Numeric Constants	36
Nonnumeric Constants	37
Variables	37
Modes of Computer Operations	38
Language Mode	38
EXTEND Mode	38
NOEXTEND Mode-PDP Only	39
Immediate Mode	39
Editing Mode	39
Basic Symbols	39
Arithmetic Symbols	39
Logic Symbols	40
Delimiters	40
Special Characters	40
Statement Numbers	41
Reserved Words	41
The BASIC Statement	41
Remark Statements	41
Statement Continuation	42
Exercises	43

### **4 Structured Programming**

**45**

Programming Structures	45
Sequence Structure	45
Selection Structure	46
Iteration Structure	47
Structure Combinations	47
Pseudocode	47
Top-Down Design	50
Structure Charts	50
Standards	53
Common Errors	53
Exercises	54



**SECTION 2****Beginning BASIC Programming 55****5 Simple Assignment and Output 56**

- Arithmetic Expressions 56
- The LET Statement 58
- The END Statement 60
- The PRINT Statement 60
  - Use of the Comma 61
  - Use of the Semicolon 62
  - Use of the TAB Option 63
  - Suppressing Vertical Spacing 64
  - Skipping Data Fields 64
- Common Errors 66
- Sample Programs 66
- Exercises 72

**6 Simple Data Entry 74**

- The READ/DATA Statements 74
- The INPUT Statement 76
- Common Errors 78
- Sample Programs 79
- Exercises 92

**7 Simple Transfer and Conditionals 95**

- The GOTO Statement 95
- The IF Statement 96
- Common Errors 98
- Sample Programs 99
- Exercises 116

**SECTION 3****Intermediate BASIC Programming 119****8 Intermediate Assignment, Output, and Data Entry 120**

- The LET Statement 120
- The PRINT USING Statement 120
  - Numeric Fields 121
  - Nonnumeric Fields 128
- Quotes 132
- The RESTORE Statement 132

The INPUT Statement	133	
Internal Function Subprograms	135	
Trigonometric Functions	137	
Integer Functions	137	
Random Number Function	137	
The Sign Function	139	
Miscellaneous Functions	139	
Common Errors	142	
Sample Programs	142	
Exercises	155	
<b>9 Intermediate Conditionals and Subroutines</b>		<b>157</b>
Conditionals	157	
Comparisons	157	
The IF Statement	157	
Case Structure-VAX Only	167	
Subroutines	170	
GOSUB/RETURN	170	
GOSUB/WHILE	171	
GOSUB/UNTIL	172	
Nesting Subroutines	172	
The ON/GOSUB Statement	173	
Common Errors	174	
Sample Programs	176	
Exercises	187	
<b>10 Intermediate Looping</b>		<b>191</b>
The FOR/NEXT Statements	192	
The WHILE/NEXT Statements	197	
The UNTIL/NEXT Statements	198	
Common Errors	200	
Sample Programs	201	
Exercises	228	
<b>11 Table Processing</b>		<b>232</b>
One-Dimensional Arrays	232	
Subscripts	233	
The DIM Statement	234	
Storing Data	236	
Accessing Arrays	238	
Outputting Data	239	
Additional Examples	239	
Two-Dimensional Arrays	242	
Subscripts	243	

The DIM Statement	243
Storing Data	244
Accessing Arrays	247
Outputting Data	248
Additional Examples	249
Common Errors	251
Sample Programs	252
Exercises	270

## **12 Modifiers** **277**

The WHILE Modifier	277
The UNTIL Modifier	279
The FOR Modifier	281
The IF Modifier	283
The UNLESS Modifier	283
Multiple Modifiers	284
Common Errors	285
Sample Programs	286
Exercises	296

## **SECTION 4**

## **Advanced BASIC Programming** **299**

### **13 Advanced Assignment and Output** **300**

User-Defined Functions	300
Multiple-Line Functions	303
String Functions for Extracting Substrings	307
String Functions for Searching and Linking	313
The CHANGE Statement	316
String Functions for Conversion	317
String Functions for Time and Date	320
Common Errors	322
Sample Programs	324
Exercises	339

### **14 Matrix Operations** **342**

Entering Values in a Matrix	342
The MAT READ Statement	342
The MAT INPUT Statement	344
Displaying Values in a Matrix	345
The MAT PRINT Statement	345
The MAT Initialization Statement	347

Matrix Calculations	348
Assignment	349
Addition and Subtraction	349
Multiplication	351
Matrix Functions	351
Inverse of a Matrix	351
Determinant of a Matrix	355
Transposition of a Matrix	355
Common Errors	357
Sample Programs	358
Exercises	366
<b>15 Advanced Data Entry and File Processing</b>	<b>369</b>
Types of External Input/Output	370
Terminal Format	371
Building a Data File in Terminal Format	371
Processing Concepts with Terminal Format Files	374
Processing Statements—Terminal Format Files	374
File Creation Using a BASIC Program	377
Sequential File Processing—Data Image File	379
Sequential File Processing—Fixed-Length Record File	379
End-of-File Conditions—The ON ERROR Statement	379
Sequential Updating	386
Virtual Array Storage	389
Creating a Virtual Array File	390
Listing a Virtual Array File	392
Random Processing of a Virtual Array File	392
Block Input/Output	397
Creating Files Using Block Input/Output	397
Listing Files Using Block Input/Output	402
Random Processing Using Block Input/Output	405
Common Errors	411
Exercises	411
<b>Appendix A Reserved Words</b>	<b>415</b>
<b>Appendix B ASCII Character Codes</b>	<b>420</b>
<b>Appendix C Error Messages</b>	<b>423</b>
<b>Appendix D Recoverable Error Processing</b>	<b>433</b>
<b>Appendix E EDT Text Editor</b>	<b>436</b>
<b>Index</b>	<b>453</b>

***SECTION 1***

---

***Introduction  
to the BASIC  
Programming  
Language***

---

# CHAPTER 1

---

## *Introduction to Computers and the BASIC Language*

---

### Functions of Computers

---

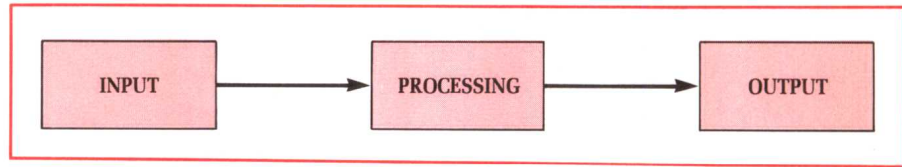
Computers are powerful tools that are used by many different types of people in many different types of organizations for many different reasons. They are used by scientists to launch and control space vehicles, by pollsters to predict outcomes of elections, by teachers to educate students, by businesses to determine net profit, and by homemakers to balance the family budget.

Computers also come in various sizes, from a large, multimillion-dollar system used by NASA that may have been manufactured by IBM or Control Data Corporation to a small personal computer used by an individual and manufactured by Apple or Radio Shack.

Regardless of the size or use of a particular computer, they all have the same essential functions, which are performed by the same types of functional units or components.

In almost any computer application, three essential functions are performed: **input**, **processing**, and **output**. The purpose of the input function is to provide a means of entering data into the computer, and the purpose of the processing function is to manipulate the data. The output function provides a means of displaying or storing the results of the processing function, which thereby generates information. This relationship is shown in Figure 1.1.

FIGURE 1.1 The Data Processing Cycle



In order for the computer to accomplish these three functions, it must have both hardware and software. **Hardware** is the physical equipment that makes up the computer system, and **software** comprises the computer programs that direct the activities of the computer. Software is divided into two types: control software and applications software. **Control software** is written to direct the activities of the computer, such as the actual movement of data from one device to another, while **applications software** is written to solve a particular problem, such as the writing of payroll checks. Control software, normally called the operating system, is usually written by the computer manufacturer. Applications software is written in a computer language such as BASIC. Such software is necessary to tell the computer hardware what to do and in what sequence to perform operations. The main purpose of this text is to present to the reader the means to use a computer to solve problems using the BASIC language.

### Processing Tasks

---

Software is written and entered into the computer to perform specific tasks. The tasks, regardless of the application, may include **calculating, storing, reproducing, summarizing, classifying, and sorting** of data. These tasks are depicted in Figure 1.2.

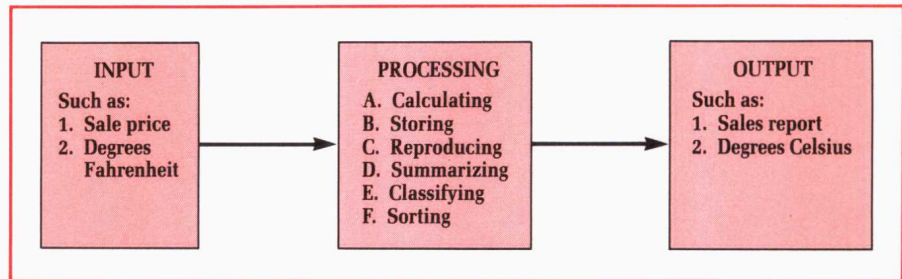
While the input and output functions are performed by devices such as a card reader, a printer, or a terminal, the processing is performed by the central processing unit (CPU) and one or more external devices, such as a magnetic disk. The **central processing unit** is comprised of three parts: the **control section**, the **arithmetic/logic unit**, and **temporary or main storage**. The external devices are used for permanent or auxiliary storage. The relationship among these components is illustrated in Figure 1.3.

### Central Processing Unit (CPU)

---

The **control section** of the CPU directs the activities of the entire computer system. Examples of activities controlled are the movement of data and/or programs from an input device into temporary storage

FIGURE 1.2 The Data Processing Tasks

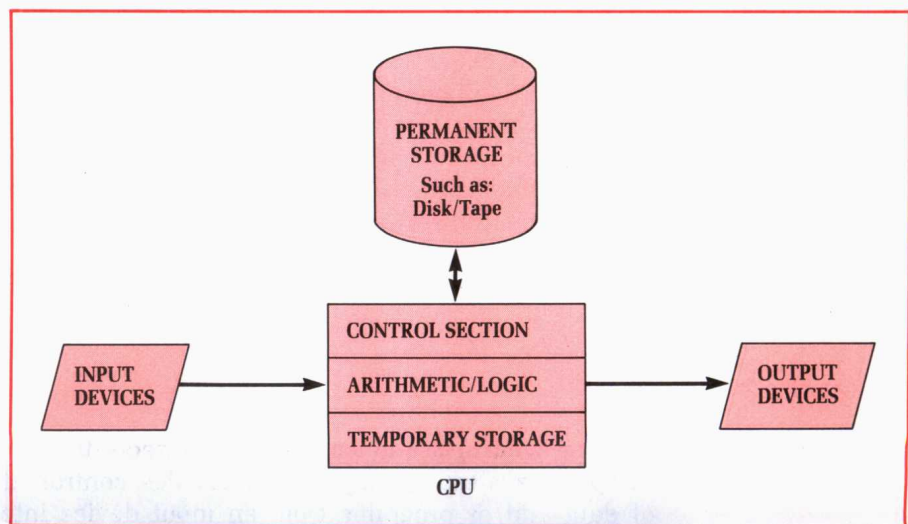


and movement of data and/or programs from temporary to permanent storage.

The **arithmetic/logic unit** performs mathematical functions, such as multiplication and rounding, and gives the computer its logic capabilities, such as comparing two values or changing the sequence of computer operations.

**Temporary storage**, also called **main** or **primary** storage, allows the storage of data and/or programs that are immediately available to the computer system. Main storage is considered to be temporary because of its limited capacity relative to auxiliary storage and because most main storage is volatile in nature. Volatile storage is subject to data loss when a power outage occurs or when the computer is intentionally turned off.

FIGURE 1.3 Data Processing Hardware





## Input, Output, and Storage Devices

---

A brief discussion of some of the different devices that may be used with a computer system is in order. While this discussion is not intended to be an exhaustive list or description, it will show the types of devices that are available on systems like Digital Equipment Corporation's (DEC) VAX or PDP systems. The VAX and PDP systems are widely used mini-computers that are representative of several manufacturers' systems.

### *Terminals*

**Terminals** are one of the major devices used to enter programs and/or data into a computer system and to accept output from the system. They allow for input into a computer via a typewriterlike keyboard; they will accept output from a computer primarily by means of a **video display terminal (VDT)** or a **teleprinter**. A VDT looks like a television screen and will display output on the screen. The teleprinter accepts output by printing onto computer paper. Both of these types of terminals are connected to the central processing unit (CPU) by means of some form of communications link, such as a telephone hookup or a direct cable.

### *Line Printers*

**Line printers** are available to output large amounts of data at high speed. They are available in different sizes, prices, and technologies. Their advantage over a teleprinter is that larger volumes of reports can be produced at high speeds.

### *Magnetic Tape*

**Magnetic tape** is a medium for permanent storage that provides large capacity and high speed of operation. It is also referred to as **auxiliary** or **secondary storage**. Data are stored electronically on the surface of the tape by magnetizing spots on the tape. Because of its high capacity and speed, magnetic tape has become a popular device for low-cost, sequential storage of data. Its major disadvantage is that access to the data is sequential; that is, *one cannot locate a specific item of information without reading the items located before the desired item*. **Sequential processing** requires that the data be in some sequence, either alphabetical or numerical. For instance, if accounts receivable data were in alphabetical order and the programmers wanted to find information on someone whose name started with an *S*, each and every person's data before the one desired would have to be read first.

### *Magnetic Disk*

**Magnetic disks** are permanent storage devices that allow direct access to any location on the disk. When this technique is used, it is called **direct** or **random processing**. Direct access to the data makes it unnecessary to search an entire disk, as is necessary for magnetic tape. Magnetic disks also allow sequential access.

There are two types of disk media: a **hard disk** and a flexible or