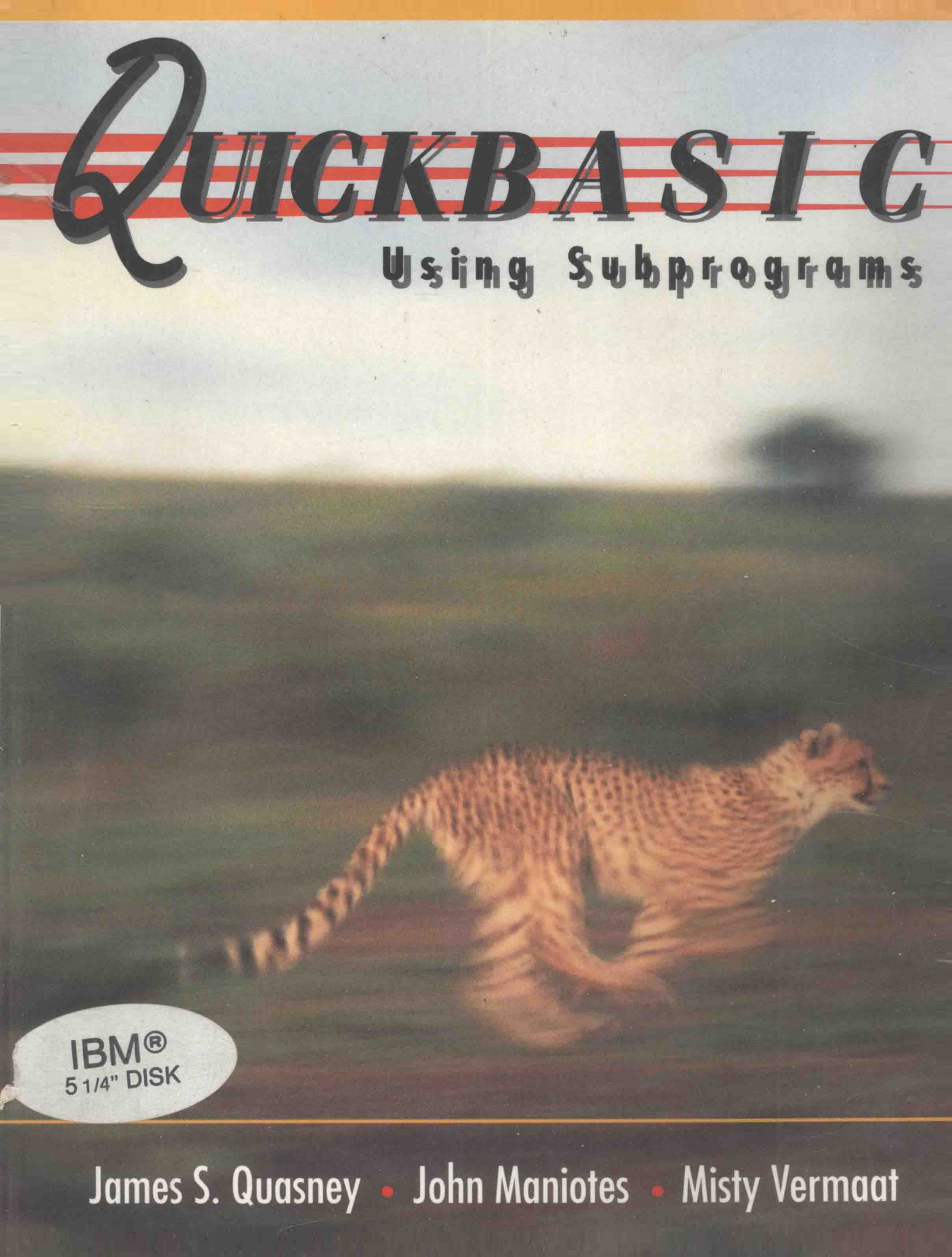


QUICKBASIC

Using Subprograms

A cheetah is shown in profile, running from left to right across a savanna landscape. The cheetah's body is in motion, with its legs extended and its tail slightly curved. The background consists of a flat, greenish-brown plain under a pale, hazy sky. A single, dark, silhouetted tree is visible in the distance on the right side of the frame.

IBM®
5 1/4" DISK

James S. Quasney • John Maniotes • Misty Vermaat

QuickBASIC Using Subprograms

James S. Quasney
John Maniotes
Misty Vermaat
Purdue University Calumet

boyd & fraser publishing company

DEDICATION

*To the past and present students, faculty, and staff of
Purdue University Calumet.*

Credits:

Acquisitions Editor: James H. Edwards

Production Editor: Barbara Worth

Director of Production: Becky Herrington

Manufacturing Coordinator: Karen Truman

Interior Design: Becky Herrington

Cover Design: Hannus Design Associates

Cover Photography: Chris Harvey, Tony Stone WorldWide

Typesetting and Composition: Huntington & Black Typography

© 1993 by boyd & fraser publishing company
A Division of South-Western Publishing Company
One Corporate Place • Ferncroft Village
Danvers, Massachusetts

All rights reserved. No part of this work may be reproduced or used in any form or by any means — graphic, electronic, or mechanical, including photocopying, recording, taping, or information and retrieval systems — without written permission from the publisher.

Names of all products mentioned herein are used for identification purposes only and may be trademarks and/or registered trademarks of their respective owners. South-Western Publishing Company and boyd & fraser publishing company disclaim any affiliation, association, or connection with, or sponsorship or endorsement by such owners.

Manufactured in the United States of America

Library of Congress Cataloging-in-Publication Data

Quasney, James S.

QuickBASIC using subprograms / James S. Quasney, John Maniotes,
Misty Vermaat.

p. cm.

Includes index.

ISBN 0-87709-010-6

1. BASIC (Computer program language) 2. Microsoft QuickBASIC.

I. Maniotes, John. II. Vermaat, Misty. III. Title.

Qa76.73.B3Q3754 1993

005.26'2--dc20

92-30589

CIP

ISBN 0-87709-010-6

1 2 3 4 5 6 7 8 9 DH 5 4 3 2

Preface

OBJECTIVES OF THIS BOOK

This book was developed specifically for an introductory computer programming course that utilizes MS-DOS QBASIC or the commercial version of Microsoft QuickBASIC on an IBM PC, IBM PS/1, IBM PS/2, or compatible system. The objectives of this book are as follows:

- To acquaint the reader with the proper and correct way to design and write high-quality programs using independent subprograms. The top-down approach, structured programming, and modern programming practices are emphasized early and are consistently used throughout the book.
- To teach the fundamentals of the QuickBASIC programming language.
- To teach good problem-solving techniques that can be used in advanced computing and information-processing courses.
- To emphasize interactive applications and menu-driven programs, the most popular type of programming in today's world.
- To develop an exercise-oriented approach that allows the reader to learn by example.
- To use practical problems to illustrate the applications of computers.
- To encourage independent study, and help those who are working alone on their own personal computer systems.

WHY USE QuickBASIC RATHER THAN BASICA?

QuickBASIC (or MS-DOS QBasic) offers a superb environment in which to learn programming. It has several important advantages over BASICA that make it indispensable for instructors who want to teach their students the correct, structured programming practices right from the start. The major advantages of QuickBASIC over BASICA are as follows:

- QuickBASIC offers subprograms and functions as an alternative to subroutines for modularizing programs.
- The block IF statement allows for multiple lines of code in the THEN and ELSE clauses.
- The SELECT CASE statement can be used to implement the Case structure without the use of the ON GOSUB statement or complex, nested IF statements.
- The DO WHILE...LOOP and DO...LOOP UNTIL statements allow students to implement cleanly the Do-While and Do-Until structures.
- The QuickBASIC editor is powerful, and yet easy to use. The operating environment, which includes windows, pull-down command menus, and the use of a mouse, helps simplify the tasks of entering, modifying, and executing programs. The QuickBASIC interpreter checks each line entered for correct syntax, formats the line, and changes the line to executable form if the syntax is correct.
- The full-screen editor allows the user to insert lines, delete lines, and change lines at the desired location on the screen, thus eliminating the need for line numbers.
- Complete mouse capabilities allow the student to easily navigate through the QuickBASIC operating environment without having to remember keystroke sequences.
- QuickBASIC uses dialog boxes with buttons whenever it needs additional information from the programmer. These dialog boxes make it easier for students to respond to requests made by QuickBASIC.
- Improved on-line debugging tools, that go far beyond the TRON and TROFF statements, are available to the student confronted with baffling logic errors.
- The QB Advisor offers a complete on-line help facility. The QB Advisor is a context-sensitive, electronic help system with instant cross-referencing that answers students' questions as fast as they can click the mouse or press the F1 key. The QB Advisor also provides programming examples that can be copied into the current program.

LEVEL OF INSTRUCTION

No previous experience with a computer is assumed, and no mathematics beyond the high school freshman level is required. The book is written specifically for the student with average ability, for whom continuity, simplicity, and practicality are characteristics we consider essential. Numerous insights, based on the authors' sixty cumulative years of experience in teaching and consulting in the field of computer information systems, are implicit throughout the book. For the past twenty years, one of us has taught introductory programming courses using a dialect of BASIC, the latest being QuickBASIC.

FUNDAMENTAL TOPICS ARE PRESENTED IN DETAIL

Besides introducing students to the correct way to design and write programs by means of structured and top-down techniques, the book presents fundamental topics concerning computers and programming which should be covered in any introductory programming class. These include the stored program concept; getting acquainted with the computer; editing programs; input/output operations; variables and constants; simple and complex computations; the use of functions and subprograms; decision making; the use of counters and running totals; rounding and truncation; looping and end-of-file tests; counter-controlled loops; the use of relational and logical operators; string manipulation; and graphics and sound. Other essential topics include data validation; control breaks; paging reports; table processing; sequence checking; selection; searching; matching; merging; sorting; file processing; and the differences between batch and interactive applications. Every one of these topics is covered in detail in this book.

DISTINGUISHING FEATURES

The distinguishing features of this book include the following:

A Proven Book

This book has evolved over the past twenty years and is based on the authors' seven prior books on BASIC programming. Many instructors and students who have used our books have shared with us their comments and suggestions for improvement as new programming techniques have been developed. They have done much to shape the contents of this book, which reflects modern programming practices using independent subprograms.

Early Presentation of the Top-Down (Modular) Approach and the Structured Programming Approach

Students are introduced to the top-down approach early, before they learn about looping and decision making. By the time they get to the larger and more complex programs, they are solving problems top-down by habit.

To implement the top-down approach, this book consistently uses independent subprograms. That is, the CALL and SUB statements are used rather than the GOSUB and RETURN statements. In Chapter 4, the student is also introduced to important design concepts, including high-level design, detailed design, and the use of stubs. These design concepts are then used throughout the book. Hence, the student is introduced early to the proper and correct way to design and code a program top down.

Particular attention is given to designing proper programs by means of the three logic structures of structured programming: Sequence; Selection (If-Then-Else and Case); and Repetition (Do-While and Do-Until). A disciplined method for implementing the structured design is adhered to throughout the book.

Early and Complete Coverage of File Processing

Complete coverage of sequential, random, and simulated-indexed files provides the reader with knowledge that is central to a real programming environment. Topics include creating all three types of files; file maintenance (matching and merging operations); and an information retrieval system that features simulated-indexed files. Sequential file processing is covered immediately following the presentation of the top-down approach and structured programming.

Student Diskette

The *Student Diskette* that accompanies this book includes all the executable programs and data files presented in the text. Students can use the program and data files on this diskette for the following:

- To step through the **PC Hands-On Exercises** at the end of each chapter
- To select a program similar to their solution for a programming assignment (this will save keying time)
- To experiment on their own with developing alternative solutions to the programming case studies presented in the text
- To access data files required in the programming assignments
- To store their solutions to programming assignments

Program names on the diskette are in the form of PRGc-n, where c represents the chapter number and n represents the program number. For example, PRG2-8 refers to the eighth program presented in Chapter 2. Data file names correspond to the names used in the text.

BASIC Programming Problems with Sample Input and Output

Over 60 challenging, field-tested BASIC Programming Problems are included at the end of the chapters. Each of the problems includes a statement of purpose, a problem statement, sample input data, and the corresponding output results. Solutions to these problems are given in the *Instructor's Manual and Answer Book* and on the *Instructor Diskette*.

Interactive Applications (Menu-Driven Programs)

Although examples of batch processing are presented, the primary emphasis is on interactive processing. The reader is introduced to the INPUT, PRINT, and CLS (Clear Screen) statements early in Chapter 2. The LOCATE statement is presented in Chapter 5 and thereafter is used extensively to build screens. Several menu-driven programs are illustrated to familiarize the reader with the type of programming that is proliferating today.

Emphasis on the Program Development Life Cycle

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

Emphasis on Fundamentals and Style

Heavy emphasis is placed on the fundamentals of producing well-written and readable programs. A disciplined style is consistently used in all program examples. Thorough documentation and indentation standards illuminate the implementation of the Selection and Repetition logic structures. The programming and style tips recommended throughout the book are summarized in Appendix C.

Summary of the QuickBASIC Language on a Reference Card

A summary of the statements, functions, special keys, operators, and reserved words can be found on a reference card at the back of the book. This summary is invaluable to the beginning student as a quick reference piece.

Presentation of Programming Case Studies

This book contains 25 completely solved and annotated case studies, illuminating the use of QuickBASIC and computer programming in the real world. Emphasis is placed on problem analysis, program design, and an in-depth discussion of the program solution. The program solutions to these programming case studies, as well as all other programs found throughout the book, are on the accompanying *Student Diskette*.

Program Design Aids

The authors recognize top-down charts and flowcharting as excellent pedagogical aids and as the tools of an analyst or programmer. Hence, many of the programming case studies include both top-down charts and program flowcharts to demonstrate programming style, design, and documentation.

Debugging Techniques and Programming Tips

A characteristic of a good programmer is that he or she has confidence that a program will work the first time it is executed. This confidence implies that careful attention has been given to the design and that the design has been fully tested. Still, errors do occur; and when they do, they must be corrected. Throughout this book, especially in Appendix C, efficient methods for locating and correcting errors are introduced using the QuickBASIC debugger. Tracing, as well as other debugging techniques, is discussed in detail. The sections in Appendix C which deal with programming tips and style tips serve as excellent references, facilitating the writing of efficient, readable code.

Applications-Oriented Approach

More than 150 QuickBASIC programs, illustrating a wide range of practical applications, along with many partial programs, are used to introduce specific statements and the proper and correct way to write programs.

Emphasis on Data Validation

The reliability of a thoroughly tested program cannot be guaranteed once it is turned over to a user. Most abnormal terminations in a production environment are due to user errors rather than programmer errors. This is especially true for programs that interact with the user or are executed on personal computers. Good programmers will attempt to trap as many user errors as possible. This book pays particular attention to the illustration of various data validation methods for ensuring that incoming data is reasonable or within limits.

What You Should Know

Each chapter contains a succinct, list-formatted review entitled What You Should Know, which reinforces key concepts and computer information system terminology.

Test Your BASIC Skills

A set of short-answer exercises identified as Test Your BASIC Skills appears at the end of each chapter. More than 200 problems, many of which are complete programs, are included for practice. Through the use of these exercises, the student can master the concepts presented, and instructors are afforded a valuable diagnostic tool. Answers for the even-numbered Test Your BASIC Skills exercises are available to the students in Appendix E. Answers to the odd-numbered exercises can be found in the *Instructor's Manual and Answer Book*.

Graphics and Sound

Chapter 11 covers all the graphics statements and functions in QuickBASIC that are central to understanding what can be done with graphics on the PC. The topics provide the student with knowledge of how to create, change, display, and store graphic designs and animation sequences. Furthermore, the necessary sound and music statements are discussed and are applied to various applications.

Additional PC Information

Besides a general introduction to personal computers in Chapter 1, Appendix D includes diskette formatting and operating instructions for the PC, and a list of popular magazines, newspapers, and manuals to help keep the student abreast of the new developments in the computer field.

ANCILLARY MATERIALS

A comprehensive instructor's support package accompanies *QuickBASIC Using Subprograms*. These ancillaries are available upon request from the publisher.

Instructor's Manual and Answer Book

The *Instructor's Manual and Answer Book* includes the following:

- Lecture outlines for each chapter
- Transparency masters from each chapter of the text
- Chapter-by-chapter objectives, teaching suggestions, and vocabulary lists
- Answers to the odd-numbered Test Your BASIC Skills exercises
- Program solutions to the more than 60 programming assignments in the book
- Test bank, including true/false, short-answer, fill-in, and multiple-choice questions for quizzes and tests

An *Instructor's Resource Diskette* is also available that includes the solutions to the more than 60 programming assignments found at the end of Chapters 2 through 11.

MicroSWAT III

MicroSWAT III, a computerized test-generating system, is available free to adopters of this textbook. It includes all of the questions from the *Instructor's Manual and Answer Book*. MicroSWAT III is an easy-to-use, menu-driven package that provides instructors with testing flexibility and allows customizing of testing documents. For example, a user of MicroSWAT III can enter his or her own questions and can generate review sheets and answer keys. MicroSWAT III will run on any IBM PC, IBM PS/1, IBM PS/2, or IBM-compatible system with a diskette drive, or a diskette drive and a hard disk.

ACKNOWLEDGMENTS

We would like to thank and express our appreciation to the many fine and talented individuals who have contributed to the success of this book. We were fortunate to have a group of reviewers whose critical evaluations of our first seven BASIC books, *Standard BASIC Programming*, *BASIC Fundamentals and Style*, *Complete BASIC for the Short Course*, *Apple-soft BASIC Fundamentals and Style*, *Structured BASIC Fundamentals and Style for the IBM PC and Compatibles*, *Structured Microsoft BASIC: Essentials for Business*, and *QuickBASIC Fundamentals and Style*, were of great value during the preparation of these books. Special thanks again go to the following:

Dory Lyn Anderson, Saint Cloud Community College
William Bailey, Casper College
Chester Bogosta, Saint Leo College
David Bradbard, Auburn University
John J. Couture, San Diego City College
Louise Darcey, Texas A&M University
I. Englander, Bentley College
George Fowler, Texas A&M University
John T. Gorgone, Bentley College
James N. Haag, University of San Francisco
Linda Kosteba, Purdue University Calumet
Riki Kuchek, Orange Coast College
Jerry Lameiro, Colorado State University
Diane Larson, Purdue University Calumet
James Larson, Homewood Flossmoor High School
Marilyn Markowitz, Purdue University Calumet
John Monroe, Roberts Wesleyan College
Donald L. Muench, St. John Fisher College
Leroy Robbins, Bee County College
John Ross, Indiana University at Kokomo
R. Waldo Roth, Taylor University
Al Schroeder, Richland College
Syed Shahabuddin, Central Michigan University
Sumit Sircar, University of Texas at Arlington
Dave Talsky, University of Wisconsin–Milwaukee
Michael Walton, Miami–Dade Community College North
Mick Watterson, Drake University
Charles M. Williams, Georgia State University

The instructional staff of the Information Systems and Computer Programming Department of Purdue University Calumet provided many helpful comments and suggestions, and to them we extend our sincere thanks. Special thanks to Sam A. Maniotes for his contributions to the problem statements and their solutions of Basic Programming Problems 11.6 to 11.9 and Exercises 11.15 to 11.19 involving graphics, animation, and music. Special thanks also goes to Debbie Fansler for testing all the programs on the student diskette and instructor diskette to ensure their correctness.

No book is possible without the motivation and support of an editorial staff. Therefore, our final acknowledgment and greatest appreciation are reserved for the following at boyd & fraser: Thomas K. Walker, publisher and valued friend, for his unfaltering support; James H. Edwards, acquisitions editor, for the opportunity to write this book; Donna Walker, a prized friend, for her constant encouragement; Barbara Worth, production editor, for her assistance; Rosanne Coit, editorial assistant, for her commitment to the project; Becky Herrington, director of production, whose creative talents show up throughout the book; and finally, special praise for Ginny Harvey, for her outstanding copy-edit work.

Hammond, Indiana
 January 1993

James S. Quasney
 John Maniotes
 Misty Vermaat

Notes to the Student

A few things to help you get going:

1. The first occurrence of a computer or programming term in this book is printed in **bold**. Its definition can be found in the same or next sentence.
2. The line numbers that appear to the left of program lines throughout this book are not part of the programs. QuickBASIC does not require line numbers. Their appearance is strictly for reference purposes during the discussion of the program. Beginning in Chapter 5, these line numbers also appear near symbols in top-down charts and program flowcharts in order to show their relationship to the corresponding program.
3. Each chapter ends with an important, useful review section called What You Should Know.
4. The answers to all the even-numbered Test Your BASIC Skills questions are in Appendix E.
5. A convenient, fully detailed language reference for QuickBASIC is never farther away than your F1 key or right Mouse button. Use these context-sensitive help keys whenever you have a question or want to learn more about the item nearest your cursor. This works for BASIC keywords, menu commands, error messages, dialog boxes, and just about anything else you can point to.
6. All the executable programs in the text are on the *Student Diskette* that accompanies this book. The programs on the *Student Diskette* which correspond to those in the text begin with the prefix PRG, followed by the chapter and program numbers. For example, PRG2-8 refers to the eighth executable program in Chapter 2.

Each chapter-ending Test Your BASIC Skills section includes several PC Hands-On Exercises that utilize the programs on the *Student Diskette*. Follow the directions and load, modify, and execute the programs. These short exercises will help you understand the significance of various QuickBASIC statements and how slight modifications to a program can affect the results.

You will also find the programs on the *Student Diskette* helpful when you are solving assigned programming problems. These programs can be retrieved from the diskette, and statements can be added, modified, or deleted to arrive at a solution. Most of the programming exercises in this book suggest which program should be loaded from the *Student Diskette* and modified to develop a solution.
7. An easy-to-use reference card at the back of this book contains a summary of the QuickBASIC statements, functions, special keys, operators, limits, and reserved words.

Contents

PREFACE	ix
NOTES TO THE STUDENT	xv
LIST OF PROGRAMMING CASE STUDIES	xvi

1 COMPUTERS AND PROBLEM SOLVING: AN INTRODUCTION

1.1 What Is a Computer?	1
1.2 Computer Hardware	2
1.3 The PC and PS Family	4
1.4 The Stored Program Concept	9
1.5 Computer Software	9
1.6 Problem Solving and Program Development	12
1.7 Additional Information on Personal Computers	16
1.8 What You Should Know	16
1.9 Test Your BASIC Skills	17
1.10 PC Hands-on Exercises	20

2 QUICKBASIC: AN INTRODUCTION

2.1 Creating A QuickBASIC Program	21
2.2 The INPUT Statement	24
2.3 The PRINT and CLS Statements	27
2.4 Coding and Documenting	29
2.5 Getting Acquainted with the QB Operating Environment	32
2.6 Editing QuickBASIC Programs	37
2.7 Executing Programs and Hard-Copy Output	39
2.8 Saving, Loading, and Erasing Programs	42
2.9 The QB Advisor On-Line Help System	44
2.10 A Guide to Writing Your First Program	46
2.11 What You Should Know	48
2.12 Test Your BASIC Skills	50
2.13 BASIC Programming Problems	53

3 CALCULATIONS, STRINGS, AND THE PRINT STATEMENT

3.1 Introduction	57
3.2 Constants	58
3.3 Variables	62
3.4 The LET Statement	65
3.5 Expressions	67
3.6 The PRINT Statement	76
3.7 What You Should Know	81
3.8 Test Your BASIC Skills	83
3.9 BASIC Programming Problems	86

4 THE TOP-DOWN APPROACH TO PROBLEM SOLVING USING SUBPROGRAMS

4.1	Introduction	91
4.2	Top-Down Design	92
4.3	Top-Down Programming Using Subprograms	94
4.4	Using the QuickBASIC Editor to Enter Subprograms	98
4.5	Designing the Logic for Subprograms	103
4.6	What You Should Know	112
4.7	Test Your BASIC Skills	114
4.8	BASIC Programming Problems	116

5 LOOPING AND INPUT/OUTPUT

5.1	Introduction	121
5.2	The DO and LOOP Statements	125
5.3	The READ, DATA, and RESTORE Statements	129
5.4	The PRINT USING Statement for Formatted Output	133
5.5	The LOCATE Statement	148
5.6	What You Should Know	154
5.7	Test Your BASIC Skills	156
5.8	BASIC Programming Problems	159

6 STRUCTURED PROGRAMMING AND MENU-DRIVEN PROGRAMS

6.1	Structured Programming	165
6.2	The IF Statement	167
6.3	Accumulators	171
6.4	Implementing the Do-While and Do-Until Structures	176
6.5	Implementing the If-Then-Else Structure	178
6.6	Logical Operators	182
6.7	Data-Validation Techniques	188
6.8	The SELECT CASE Statement and Menu-Driven Programs	191
6.9	What You Should Know	203
6.10	Test Your BASIC Skills	205
6.11	BASIC Programming Problems	212

7 SEQUENTIAL FILES, PAGING REPORTS, AND CONTROL-BREAK PROCESSING

7.1	Introduction	221
7.2	Data Files	221
7.3	Sequential File Processing	223
7.4	Paging a Report	237
7.5	Control-Break Processing	244
7.6	What You Should Know	256
7.7	Test Your BASIC Skills	257
7.8	BASIC Programming Problems	258

8 FOR LOOPS, ARRAYS, SORTING, AND TABLE PROCESSING

8.1	Introduction	265
8.2	The FOR and NEXT Statements	265
8.3	Arrays Versus Simple Variables	276
8.4	Declaring Arrays	277
8.5	Manipulating Arrays	280
8.6	Multidimensional Arrays	287
8.7	Sorting	291
8.8	Table Processing	297
8.9	What You Should Know	310
8.10	Test Your BASIC Skills	312
8.11	BASIC Programming Problems	317

9 MORE ON STRINGS AND FUNCTIONS

9.1	Introduction	325
9.2	String Functions and Statements	325
9.3	Numeric Functions	346
9.4	User-Defined Functions	359
9.5	Independent Functions	365
9.6	Trapping Events	369
9.7	What You Should Know	372
9.8	Test Your BASIC Skills	373
9.9	BASIC Programming Problems	377

10 FILE MAINTENANCE, RANDOM FILE PROCESSING, AND SIMULATED-INDEXED FILES

10.1	Introduction	385
10.2	File Maintenance	386
10.3	Random File Processing	399
10.4	Simulated-Indexed Files	412
10.5	What You Should Know	425
10.6	Test Your BASIC Skills	426
10.7	BASIC Programming Problems	428

11 COMPUTER GRAPHICS AND SOUND

11.1	Introduction	433
11.2	Text-Mode Graphics	439
11.3	Medium-Resolution and High-Resolution Graphics	456
11.4	Sound and Music	477
11.5	What You Should Know	480
11.6	Test Your BASIC Skills	481
11.7	BASIC Programming Problems	487

APPENDIX A PROGRAM DESIGN TOOLS — FLOWCHARTS, PSEUDOCODE, NASSI-SCHNEIDERMAN CHARTS, AND WARNIER-ORR DIAGRAMS

A.1	Program Flowcharting	493
A.2	Guidelines for Preparation of Flowcharts	495
A.3	Control Structures	499
A.4	Proper Programs	500
A.5	Flowcharting Tips	500
A.6	Pseudocode	502
A.7	Nassi-Schneiderman Charts	504
A.8	Warnier-Orr Diagrams	506
A.9	Test Your BASIC Skills	507

APPENDIX B MENU COMMANDS AND WINDOWS

B.1	Selecting Commands	511
B.2	The MS-DOS QBASIC Menu Bar	512
B.3	The QuickBASIC Commercial Version Menu Bar (Full Menus Only)	518
B.4	Using Windows to Your Advantage	525

APPENDIX C QuickBASIC DEBUGGING TECHNIQUES, CHAIN STATEMENT AND PROGRAMMING TIPS

C.1	Debugging Techniques	527
C.2	Trapping User Errors (ON ERROR GOTO and RESUME)	530
C.3	Chaining	534
C.4	Programming Tips	536
C.5	Program Style Tips	539

APPENDIX D FORMATTING A DISKETTE, DOS COMMANDS, ASCII CHARACTER SET, AND PERSONAL COMPUTER LITERATURE

D.1	Formatting a Diskette	541
D.2	DOS Commands	542
D.3	QuickBASIC DOS-Like Statements	544
D.4	ASCII Character Codes	544
D.5	Personal Computer Magazines, Newspapers, and Manuals	546

APPENDIX E ANSWERS TO THE EVEN-NUMBERED TEST YOUR BASIC SKILLS EXERCISES

INDEX

QuickBASIC REFERENCE CARD

STUDENT DISKETTE

List of Programming Case Studies

<i>NO.</i>	<i>CASE STUDY</i>	<i>PAGE</i>
1	Computing an Average	11
2	Determining a Salesperson's Commission	21
3	Tailor's Calculations	57
4A	Determining the Single Discount Rate	66
4B	Determining the Single Discount Rate with a Fixed-Screen Format and an End-of-File Test	149
5	Determining the Sale Price	121
6	Determining the Accounts Receivable Balance	143
7A	Weekly Payroll and Summary Report	171
7B	Writing the Weekly Payroll and Summary Report to Auxiliary Storage	227
8	A Menu-Driven Program	195
9	Creating a Sequential File	231
10	Processing a Sequential Data File and Paging a Report	238
11A	Sales Analysis Report — Single-Level Control Break	246
11B	Sales Analysis Report — Two Levels of Control Breaks	251
12	Analysis of Monthly Sales	283
13	Deciphering a Coded Message	334
14	Validating Payment Dates	338
15	Determining the Time It Takes to Double an Investment	349
16	Guess a Number Between 1 and 100	357
17	Computer Simulation — Beat the House Roller	362
18	File Maintenance I — Adding Records by Merging Files	387
19	File Maintenance II — Deleting and Changing Records by Matching Records	393
20	Creating a Random File	405
21	Accessing Records in a Random File	409
22	Using a Simulated-Indexed File for Inventory Retrieval and Update	412
23	Logo for the Bow-Wow Dog Food Company	440
24	Horizontal Bar Graph of Monthly Sales	443
25	Animating an Inchworm Creeping Across the Screen	452

Computers and Problem Solving: An Introduction

1.1 WHAT IS A COMPUTER?

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

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

Advantages of a Computer

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

Disadvantages of a Computer

Some of the disadvantages of a computer concern obsolescence and ongoing costs for training and maintenance. Currently, computer models become technologically obsolete in a matter of a few years. Furthermore, in order for an organization's staff to derive its maximum benefits from a computer, the organization must continually invest in training and maintenance.

1.2 COMPUTER HARDWARE

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

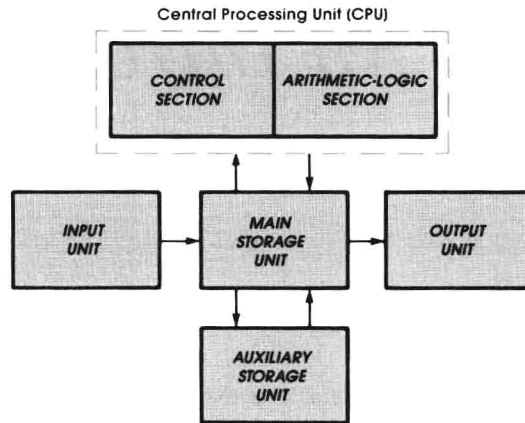


FIGURE 1.1

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

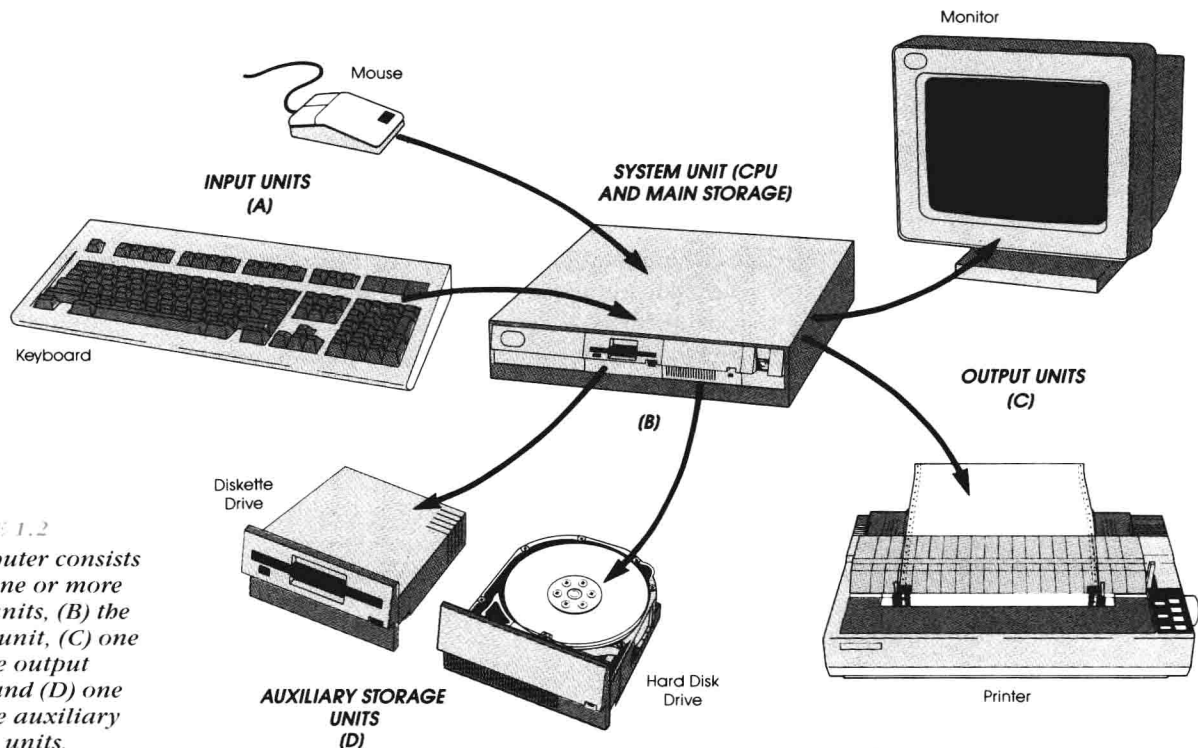


FIGURE 1.2

A computer consists of (A) one or more input units, (B) the system unit, (C) one or more output units, and (D) one or more auxiliary storage units.