

BASIC



Bijan Mashaw

BASIC

Bijan Mashaw

California State University, Hayward



MAYFIELD PUBLISHING COMPANY

To my only son, Arsheeya

Sponsoring editor: Raleigh S. Wilson, Jr.

Manuscript editor: Antonio Padial

Managing editor: Pat Herbst

Art director: Nancy Sears

Designer (interior and cover): Gary A. Head

Cover art: *Prism* (watercolor, 1966), painted by Jürgen Peters

Technical artist: Kelly Solís-Navarro

Production: Mary Forkner and Darlene Bledsoe, Publication Alternatives

Compositor: Graphic Typesetting Service

Copyright © 1985 by Bijan Mashaw

First edition

All rights reserved. No portion of this book may be reproduced in any form or by any means without written permission of the publisher.

Library of Congress Catalog Card Number: 84-062032

International Standard Book Number: 0-87484-692-7

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2

Mayfield Publishing Company

1240 Villa Street

Mountain View, California 94041

Disclaimer of Liabilities: Care has been exercised to ensure the accuracy of this book. This text, however, is meant to be used only as an instructional tool. Neither the author nor the publisher is liable for any damage incurred directly or indirectly through the use of this book.

Preface

Since I began programming in the early seventies, I have often wondered whether there could be a better and more efficient way to teach and learn programming. To try to find a better way, I studied factors that seem to increase the efficiency of learning a skill-oriented subject such as programming. The following factors are important:

1. Students learn better with a model. The model can be a figure or an example.
2. Students should practice what is being taught.
3. The instructor should emphasize problem solving, program design, and development, rather than the syntax of the language.
4. The building-block approach (using progressively difficult examples) helps students understand the process of programming.

Through years of teaching and working with students, I have found these conditions invaluable to the learning process. This book synthesizes my discoveries about the most effective ways to teach programming.

The Unique Approach of This Text

This book combines problem-solving methodology with BASIC programming language, a popular and powerful language used with many computer systems. I have tried to present carefully designed, concrete, and meaningful examples and avoid unimaginative ones. Style has been emphasized, as well as the importance of practicing programming and full-fledged program writing at an early stage in the learning process.

The method of presentation is inductive; each example has been designed to introduce programming concepts and practices. After spending a few minutes on an example, students should be able to understand and

apply the concepts, rather than merely copy them. The examples and problems are intended to encourage students to create their own applications.

This unique and innovative approach was designed specifically for readers who have had no previous programming instruction. Readers are taught to write programs from the beginning. Other features of the text include:

- Many problems and solutions to enhance the students' self-confidence and inspire new questions
- Special emphasis on program planning, problem-solving and programming techniques, structured programming, and the modular approach
- Abundant exercises and a variety of applications to challenge the students to practice what they have learned

Organization

Each section of the text starts with an example, followed by several points to illustrate the learning objective for that example. These examples are designed to help students understand the program logic, structure, and the problem-solving techniques to be used. Solved problems are presented next to reinforce the learning process and point out novice programmers' common mistakes. Finally, numerous exercises at the end of each chapter, arranged from simple to complex, allow students to practice applying the concepts they have just learned.

No knowledge of mathematics beyond basic algebra or data processing is required or assumed (except in Chapter 10, which discusses applications in several subject areas). Straightforward language is used throughout the text. Concepts are introduced gradually to make learning easier. These concepts include language syntax, problem-solving techniques, structured programming concepts, planning, the modular approach, good programming practices, and the decomposition of a complex problem into a hierarchy of simpler problems.

Topics Covered

The chapter sequence is designed to be flexible; most chapters are independent of the others. Chapter 2 covers the core material. Other chapters can be reordered to meet the needs of a specific course or presentation. For example, Section One of Chapter 11 (on file processing) can be covered right after Chapter 2. Chapter 6, which presents topics such as output editing (PRINT-USING), is not a prerequisite for any other chapter. A brief summary of the prerequisites for each chapter is shown on the next page.

<i>Chapter</i>	<i>Topics</i>	<i>Prerequisite chapter(s)</i>
1	Simple programs	—
2	Basics of output, input, and processing	—
3	Interpreter's job, program planning	2
4	IF statements, programming styles	2
5	Looping techniques	2, 4
6	Output editing, data types, and strings	2, 4, 5
7	One-dimensional arrays	2, 4, 5
8	Two-dimensional arrays and matrices	2, 4, 5, 7
9	Subroutines and functions	2*
10	Applications	2, 4, 5, 7, 8, 9
11	File processing (Section I)	2
	File processing (Section II)	2, 4, 5, 9
12	Program efficiency and structured programming	2, 3, 4, 5, 7, 9

* A few examples use the FOR-NEXT loop.

Version of BASIC

This text is not written for any specific version of BASIC, because the emphasis is on the basic techniques of programming and problem solving, rather than a specific syntax version. The text emphasizes the existence of a variety of versions, and students are expected to convert a program written in one version of BASIC to another.

All programs in the text have been tested with MBASIC and are portable to any machine, although some might require slight modification. For example, a variable's symbolic name in a program can be longer than one character, but these names were deliberately chosen so that, if all characters after the first are dropped, the program will still run with any version of BASIC. This is explained from the start; nevertheless, the programs are presented as educational vehicles and are not canned.

Alternative Ways of Using the Text

This text covers some rather broad topics, but it remains flexible enough to be used in various ways. The book can be readily used as a text for a three-hour course on BASIC. Chapters 1–5 and selected chapters can be used to present a one-hour introduction to programming. The book could

also serve as a supplementary text for a data processing course. It is also structured to be used as a self-guided text suitable for large classes or for individuals interested in learning BASIC.

ACKNOWLEDGMENTS

I would like to express my appreciation to the many individuals who encouraged me in this project. Special thanks to Jerry Weinberg for his encouragement; to Raleigh Wilson, the acquiring editor, for providing motivation; and to Tammy Lord for her patience with the word processor.

Contributions and suggestions were received from many sources. Reviewers are crucial to the development of a successful text, and I have benefited from the advice of these superb colleagues:

James J. Ball, Indiana State University

Murray Berkowitz, University of Texas, Arlington

R. Wayne Headrick, Texas A & M University

Terry Helford, Chaffey Community College

Phillip Mackey, Delaware Technical and Community College

W. Leon Pearce, Drake University

Paul W. Ross, Millersville University

Leonard C. Schwab, California State University, Hayward

Francis J. Svlich, Fresno City College

Charles J. Wertz, State University of New York, College at Buffalo

Bijan Mashaw

BASIC

Contents

Preface xiii

Chapter 1

Basics of Programming 1

Organization of the Text 3

Overview of a Computer System 4

Computer Components 5 · Types of Computers 9 · Types of Software 11

Programs and Languages 12

Writing Programs 13

Some Simple Programs 13 · Arithmetic Operators 16 · Priorities of Arithmetic Operations 20 · Variables 23 · The LET Statement 30 · Writing Better Programs 30

Running a Program 32

System Commands 33 · Microcomputer Commands: LLIST / LPRINT 35 · BASIC Immediate Mode 35

Exercises 37

Summary 39

Review Questions/Self-Test 40

Chapter 2

Input and Output 43

Constants in BASIC 46

Numeric Constants 46 · String Constants 46

Output 47

The PRINT Statement 47 · Separators in PRINT Statements 49

Input 56

The READ-DATA Statements 57 · The RESTORE Statement 63 · The INPUT Statement 63

Some Programming Techniques 65

Looping 65 · String Variables 69 · Generating Sequence Numbers 73 · Accumulating Data: Finding the Total 74 · A Complete Program 76 · An Interactive Program 78 · A Game Program 80

Program Spacing 82

Output Spacing 82 · Spaces in the Program 84

Exercises 87

Summary 91

Review Questions/Self-Test 93

Chapter 3

Program Development and Structured Programming 95

Compilation 97

Compiler 98 · Interpreter 98 · Versions of BASIC 100 · BASIC Statements 101

Error Detection 101

Syntax Errors 101 · Execution Errors 103 · Logical Errors 103 · How to Debug a Program 104

Program Planning 107

Algorithms 107 · Programming Tools 109 · Programming Cycle 118

Writing Structured Programs 121

Sequence Structure 121 · Selection Structure 122 · Iteration Structure 123

How to Write a Structured Program 123

Structured Design 123 · Readability 126 · Reliability 129 · Documentation 129

A Case—Printing a Report 130

Step 1: The Problem 130 · Step 2: Input-Output Design 131 · Step 3: Process Design 132 · Step 4: Coding 134 · Step 5: Running, Testing, and Debugging 136 · Step 6: Documentation 140

Exercises 140

Summary 143

Review Questions/Self-Test 144

Chapter 4

Decision Making, Comparing, Branching 145

The IF Statement 147

Problems Using the IF Statement 153

Finding the Largest Number 153 · Terminating a Loop 157

Programming Style and Structured IF Block 161

GOTO Style 162 · GOTO-Less Style 163 · Structured Style 164

Structured Programming 167

Compound IF Statement: Logical Operators 173

Conditional Branching 176

Games 180

Menu in a Program 181

A Case—Generating a Report 182

Step 1: The Problem 183 · Step 2: Input-Output Design 183 · Step 3: Process Design 185 · Step 4: Coding 188 · Step 5: Execution, Testing, and Debugging 189 · Step 6: Documentation 192

Exercises 194

Summary 200

Review Questions/Self-Test 202

Chapter 5

Looping 205

The FOR-NEXT Loop 207

The General Form of the FOR-NEXT Loop 210 · Loop Flowcharting 212 · Rules of a FOR-NEXT Loop 214

Nested Loop 225

The WHILE Loop 231

Writing Better Programs 232

Exercises 233

Summary 240

Review Questions/Self-Test 240

Chapter 6

Editing, Data Type, and Strings 243

Output Editing 245

PRINT USING Statement 246 · The Format 247 · Printing Strings with PRINT USING 247 · Editing Symbols 251

Data Form 257

Numeric Data 257 · Character Data 263

Control Design 274

Exercises 275

Summary 281

Review Questions/Self-Test 282

Chapter 7

One-Dimensional Arrays 285

Array Techniques 287

Subscripted Variables 290 · The Dimension Statement 291 · Reason for Arrays: An Example 294 · Arithmetic Expressions with Arrays 296 · Array Input-Output 299 · Printing Values on One Line 302 · Mirror Printing 302 · Summary of Important Points About Arrays 302

Some Examples of Arrays and Programming Techniques 304

Exercises 312

Summary 321

Review Questions/Self-Test 322

Chapter 8

Two-Dimensional Arrays 325

Array Techniques 327

Dimension Statement 328 · Arithmetic Expressions with Two-Dimensional Arrays 330 · Two-Dimensional Array Input-Output 332 · Summary of Important Points 336

Examples of Two-Dimensional Arrays and Programming Techniques 336

Matrix Operations 346

MAT Statement 346 · Matrix Input-Output 346 · Matrix Manipulation 347 · Matrix Initialization 348

Exercises 350

Summary 358

Review Questions/Self-Test 359

Chapter 9

Subprograms 361

Subroutines 363

Internal Subroutines 364 · External Subroutines 368

Functions 370

Library Functions 370 · User-Defined Functions 373 · Summary of the Rules for Using Functions 377

Importance of Modular Programming 380

Exercises 385

Summary 393

Review Questions/Self-Test 394

Chapter 10

Problem Solving, Programming Techniques, and Some Applications 397

Data Processing 399

Sorting 400 · Searching 404 · Merging 407

Statistics 409

Median 409 · Summation Notation 410 · Regression Analysis 411

Graphing Techniques 413

Plotting a Function 413 · Graphing a Histogram 415

Simulation 418

Random Numbers 418 · Generating Random Numbers 419 · Monte Carlo Simulation 421

Games 428

Management Information and Decision Support Systems 432

Exercises 434

Summary 451

Review Questions/Self-Test 452

Chapter 11

File Processing 455

File Processing—Section I 457

What is a File? 457 · Storage Media 459 · File Organization 461

Sequential File Processing 461

General Steps 461 · Creating a File 462 · Forms of the Statements 463 · Accessing a File 467

File Processing—Section II 469

Control Break Reports 469 · Sorting 472 · Merging 474 · File Maintenance 476

Random-File Processing 483

General Steps 483 · Forms of the Statements 488
Multirecords 490 · Random Accessing by an ID 493

Exercises 495

Summary 504

Review Questions/Self-Test 504

Chapter 12

Program Efficiency 509

Efficiency Factors 511

Interpreter Optimization 512 · Sparing Use of Arrays 513 · Efficient Algorithms 514 · Human Resources 517

Structured Approach 518

Structured Design 518 · Readability 519 · Documentation and Reliability 524

Appendix A

Reserved Words for Selected Systems 525

MicroSoft BASIC Reserved Words (IBM-PC, DEC Rainbow, and Others) 525

BASIC-PLUS Reserved Words 526

Radio Shack TRS-80 LEVEL II Reserved Words 527

APPLESOFT Reserved Words 528

Appendix B

Some BASIC Syntaxes for Selected Systems 529

Appendix C

True BASIC 533

Appendix D

Basic Keywords Summary 535

Index 537

Basics of Programming

1

In This Chapter:

Organization of the Text

Overview of a Computer System

Computer Components

Types of Computers

Types of Software

Programs and Languages

Writing Programs

Some Simple Programs

Arithmetic Operators

Priorities of Arithmetic Operations

Variables

The LET Statement

Writing Better Programs

Running a Program

System Commands

Microcomputer Commands:

LLIST/LPRINT

BASIC Immediate Mode

Exercises

Summary

Review Questions/Self-Test

