

**Fourth Edition**

# PASCAL

**Walter Savitch**

**Fourth Edition**

# PASCAL

AN INTRODUCTION TO  
THE ART AND SCIENCE  
OF PROGRAMMING

**Walter Savitch**

**University of California, San Diego**



**THE BENJAMIN/CUMMINGS PUBLISHING COMPANY, INC.**

Redwood City, California • Menlo Park, California  
Reading Massachusetts • New York • Don Mills, Ontario  
Workingham, U.K. • Amsterdam • Bonn • Sydney  
Singapore • Tokyo • Madrid • San Juan

---

Acquisitions Editor: J. Carter Shanklin  
Executive Editor: Dan Joraanstad  
Editorial Assistant: Melissa Standen  
Production Editor: Teri Holden  
Copyeditor: Chris Grisonich

Proofreader: Joe Ruddick  
Interior Design Modifications: Gary Head  
Cover Design: Terry Hight Design  
Marketing Manager: Mary Tudor  
Manufacturing Coordinator: Janet Weaver

Acknowledgements:

Chapter 2, opening quotation from E. W. Dijkstra, *Notes on Structured Programming*, O. J. Dahl, E. W. Dijkstra, and C. A. R. Hoare, Academic Press (1972). Reprinted by permission of the author and publisher.

Chapter 7, ending quotation from F. P. Brooks, Jr., *The Mythical Man-Month, Essays on Software Engineering*, p. 116. Addison-Wesley Publishing Company (1975). Reprinted by permission of the publisher.

Chapter 11, ending quotation from Niklaus Wirth, *Algorithms + Data Structures = Programs*, title. Prentice-Hall (1976). Reprinted by permission of the publisher.

Chapter 14, opening quotation from John R. Ross, *Constraints on Variables in Syntax*, p.i., Ph.D. dissertation, Massachusetts Institute of Technology (1967). Reprinted by permission of the author.

Chapter 14, midchapter quotation from Jorge Luis Borges, "The Garden of Forking Paths," in Jorge Luis Borges, *Selected Stories and Other Writings*, p. 25. New Directions Publishing Company (1964). Reprinted by permission of the publisher.

Chapter 15, ending quotation from B. W. Kernighan and P. J. Plauger, *The Elements of Programming Style*, 2nd ed., p. 117. McGraw-Hill Book Co. (1978). Reprinted by permission of the publisher and Bell Laboratories.

UNIX is a registered trademark in the U.S.A. and other countries, licensed exclusively through X/Open Company Ltd.

Copyright © 1995 by The Benjamin/Cummings Publishing Company, Inc.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permissions of the publisher. Printed in the United States of America.

The programs and the applications presented in this book have been included for their instructional value. They have been tested with care but are not guaranteed for any particular purpose. The publisher does not offer any warranties or representations, nor does it accept any liabilities with respect to the programs or applications.

Library of Congress Cataloging-in-Publication Data

Savitch, Walter J., 1943—

Pascal, an introduction to the art and science of programming /  
Walter Savitch. — 4th ed.  
p. cm.

Includes bibliographical references and index.

ISBN 0-8053-7458-2

1. PASCAL (Computer program language) 2. Electronic  
digital computers—Programming. I. Title.

QA76.73.P2S28 1994

005.13'3—dc20

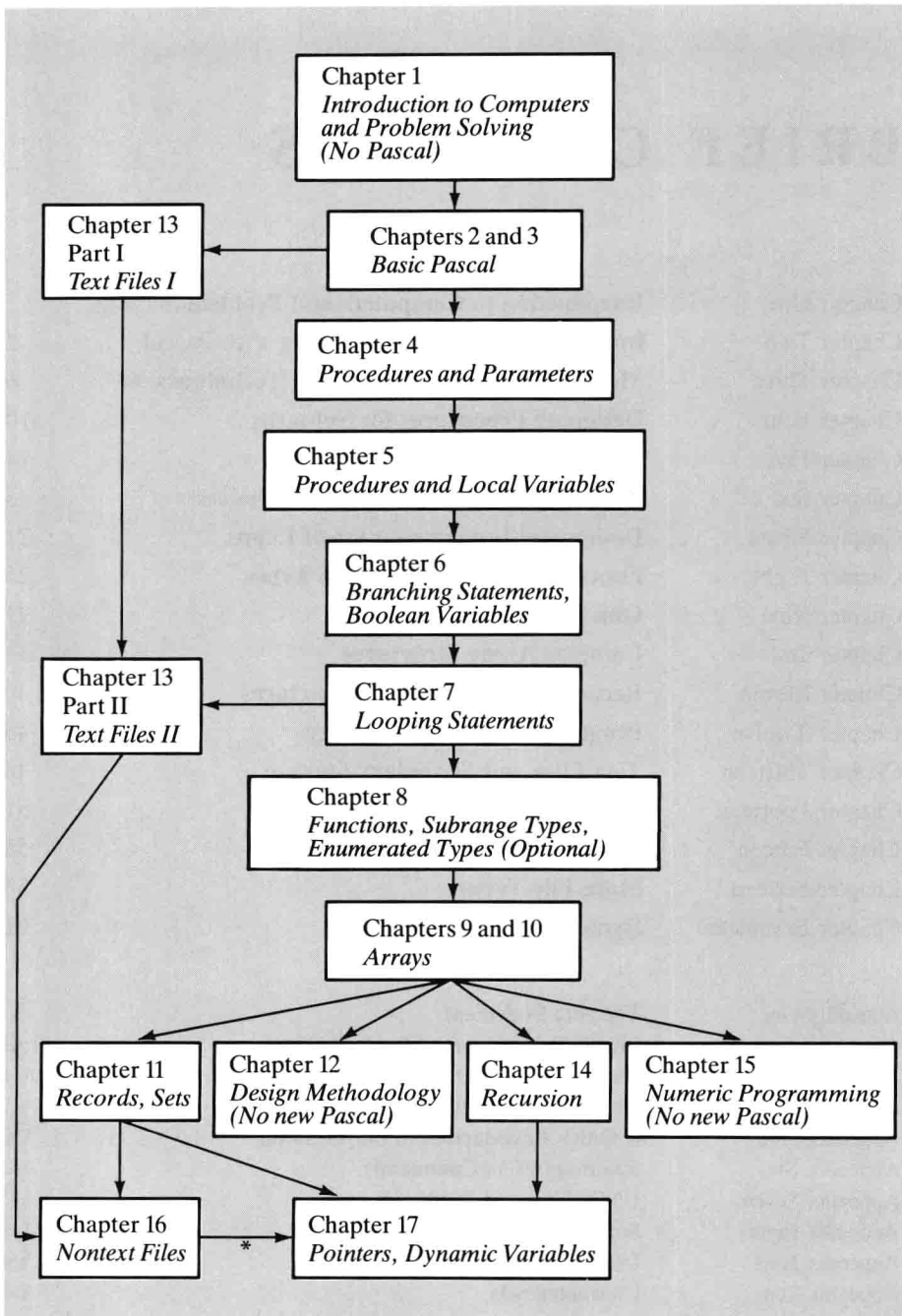
94-23006

CIP

3 4 5 6 7 8 9 10 – DOC – 02 01 00

The Benjamin/Cummings Publishing Company, Inc.  
390 Bridge Parkway  
Redwood City, California 94065

## Dependency Chart of Chapters and Location of Key Pascal Constructs



\*Two case studies in Chapter 17 use files.  
The remainder of Chapter 17 does not use files.

---

# PREFACE

This book was designed for use in introductory computer science or programming classes that use the Pascal language. This edition uses standard Pascal. (There is a separate edition designed for users of Borland's Turbo Pascal.) This book can be used for courses as short as one quarter or as long as one academic year. It includes a thorough introduction to problem solving and programming techniques as well as a complete description of standard Pascal. It assumes no knowledge of computers and no mathematics beyond high school algebra. Changes made since the last edition and some of the main features of the book are described below.

---

## Changes From the Previous Edition

This fourth edition has the same pedagogical approach as the third edition. The ordering of topics is basically the same as in the third edition. A course designed around the third edition can be carried over to this edition almost without change. However, this new edition contains additional topics that can be used to enhance a course. This new material can be introduced immediately or phased in over several different offerings of the same course. Two topics that receive extensive new coverage in this edition are abstract data types and analysis of algorithms. This new edition has a greater emphasis on abstract data types (ADTs), including a more detailed description of how ADTs can be implemented in standard Pascal. New examples of ADTs have been added at various places throughout the book. This edition also has new material on analysis of algorithm including coverage of big  $O$  notation. The coverage of big  $O$  notation is designed so that it can be introduced early, late, or even completely omitted from a course. This new edition also contains many smaller changes designed to make the book more accessible to students and more useful for instructors. The chapters on loops and on arrays have been rewritten to be more up to date. Other smaller sections throughout the book have been rewritten. We have added more material on the UNIX operating system as well as new material on an alternative Pascal environment known as Dr. Pascal. This material on particular systems is confined to appendixes so that it is truly optional.

---

## Computer Science, Not Just Programming

This textbook includes extensive coverage of Software Engineering concepts and Computer Science theory. Topics covered include the software life cycle, abstract data

types, analysis of algorithms including the big  $O$  notation, sorting and searching, program verification and loop invariants, as well as other conceptual material. A complete chapter on program design methodologies summarizes and elaborates on the software development techniques emphasized throughout this book.

---

## Early Introduction of Procedures

As with previous texts in this series, this book introduces procedures very early and presents a complete discussion of parameters as soon as procedures are introduced. Both value and variable parameters are introduced immediately after introducing the notion of a procedure.

---

## Abstract Data Types

One of the shortcomings of standard Pascal is that most standard Pascal compilers do not have built-in facilities for defining Abstract Data Types (ADTs) in a way that uses accepted principles of information hiding. Nonetheless, students in an introductory Pascal-oriented class should receive some information on this modern topic. In order to teach ADTs using standard Pascal, this book gives a detailed presentation of an easy-to-learn style for writing ADTs. This approach emphasizes the information hiding aspects of ADTs, and yet it can be used with any standard Pascal compiler. To emphasize the information hiding features of ADTs, this book gives examples showing that a program which uses an ADT can be written before the implementation of the ADT is written. Since the program cannot be run without the implementation, this book also gives careful instructions for combining an ADT and an algorithm that uses the ADT to produce a complete standard Pascal program. ADTs and examples of ADTs are presented at several points in the book, but ADTs are not presented at the cost of cutting out more traditional material. Our intention is that ADTs should be an integral part of any Pascal-oriented course. However, if desired, it is possible to skip the sections on ADTs without losing continuity of the rest of the material.

---

## Thorough Coverage of Problem Solving Techniques

The text includes complete coverage of problem solving techniques and illustrates these techniques with case studies showing the complete design process in detail from problem definition to final Pascal program.

---

## Thorough Coverage of Debugging and Programming Techniques

Thorough coverage of debugging and programming techniques is presented throughout the text. Boxed sections on common Pitfalls highlight important techniques in a compact easy-to-find and easy-to-digest way.

---

---

## Extensive Array Coverage

In addition to extensive coverage of new topics, this book includes thorough coverage of the older “bread and butter” issues of day-to-day programming. For example, two full chapters are dedicated to arrays. These chapters cover one-dimensional and multi-dimensional arrays with heavy emphasis on programming techniques such as reading data into arrays, manipulating partially filled arrays, and designing data structures using arrays. The array coverage is followed by a chapter with complete coverage of records. This permits covering records immediately after arrays (or later, if that is preferred).

---

## Extensive Coverage of Recursion

An entire chapter is dedicated to recursion. Extensive use of figures and examples makes this difficult topic accessible to beginners.

---

## Analysis of Algorithms and Big O Notation

Chapter 12 includes a new section on analysis of algorithms and big *O* notation. The material on big *O* notation is written so that it can be covered at any desired point. It can be moved forward or delayed until a later chapter. The material on big *O* notation is optional and may be completely omitted.

---

## UNIX Appendixes

Appendixes on basic UNIX file manipulation, input/output redirection, and the vi editor are included at the end of the book. This eliminates the need for a separate UNIX manual for those classes that use the UNIX system.

---

## The Dr. Pascal System

This book can be used with any system that is running a version of standard Pascal. If, however, you are still choosing a system, we have some suggestions and software that may help you with your planning. One very nice system for an introductory course is the Dr. Pascal system produced by the Visible Software Co. The Dr. Pascal system provides an editor, debugger, and a standard Pascal compiler together in one integrated environment. The Dr. Pascal system was designed specifically to be used in introductory courses. The debugger displays variable values and procedure calls in a layout that is nice for any programmer, and is particularly clear to beginning programmers. Dr. Pascal has a number of other nice little additions that are important for a beginning student, such as the ability to restart a program that has failed due to incorrect input. Versions of Dr. Pascal are available for PC, Macintosh, and VMS systems.

---

---

## Discounted and Free Pascal Interpreters

A free copy of the complete Dr. Pascal system in either a Mac or PC compatible version is available to qualified instructors upon adoption of this text while supplies last. A coupon that allows students to buy the complete Dr. Pascal system at a discounted price is included at the end of this book.

A less elaborate Pascal interpreter (produced by the same company that produces Dr. Pascal) is available free via anonymous ftp from the publishers of this book. This interpreter can be distributed free to students so that each student who owns a home computer can have her or his own Pascal interpreter. There are versions of this command line interpreter for both PC and Macintosh systems. Instructions for obtaining the interpreter via anonymous ftp are included in the section of this preface entitled Supplements and ftp.

---

## Flexibility

This book has a modular organization. Each chapter covers a topic in its entirety. For example, procedures and parameters are covered in complete detail in one chapter; all loop statements are covered in the chapter on loops; array topics are grouped together into two consecutive chapters. This makes rearranging the order in which chapters are covered easier, and it increases the value of the book as a reference source. The dependency chart at the front of this book shows the possible orders in which the chapters can be covered without losing continuity.

---

## Allows Early or Late Coverage of Text Files

The chapter on text files is divided into two parts to allow for two possibilities, either postponing the topic entirely until later in a course or briefly introducing them early and giving more detail later on.

---

## Optional Advanced Topics

Most advanced topics—such as big  $O$  notation, recursion, some software engineering topics, some numeric programming techniques, and a substantial amount of material on data structures including records, files, and pointers—are packaged into chapters that can be covered in almost any order. Alternatively, a subset of the chapters may be chosen to form a shorter course. To add even more flexibility, sections with optional topics are included throughout the book.

---



---

## Supplements and FTP

The following supporting materials are available from the publisher:

### **INSTRUCTOR'S GUIDE**

A complete chapter-by-chapter instructor's guide is available from the publisher only in a hard copy format. ISBN: 37462-0.

### **COMPUTERIZED TEST BANK**

Software for automatically generating tests corresponding to each chapter is also available to instructors. ISBN: 37464-7.

### **HARD COPY TEST BANK**

A chapter-by-chapter hard copy listing of multiple choice test questions is available to instructors. These questions are the same as the questions used by the COMPUTERIZED TEST BANK. Hard copy test bank ISBN: 37463-9.

### **PROGRAM CODE**

All of the programs in this text are available in machine-readable form. Available via anonymous FTP only. Directory bc/savitch/pascal/progcode.

### **ALGORITHMS IN ACTION**

Tutorial software for the student including lessons demonstrating algorithm behavior and programming concepts along with a collection of 21 quizzes. Available via anonymous FTP only. Directory: bc/savitch/pascal/algo-act.

### **DR. PASCAL**

An integrated programming environment including an editor, debugger, and a standard Pascal compiler published by Visible Software. A free copy of the complete Dr. Pascal system in either a Mac or PC compatible version is available to qualified instructors upon adoption of the book while supplies last. Contact your local Benjamin/Cummings Sales representative for more details.

### **COMMAND-LINE INTERPRETERS**

Command line standard Pascal interpreters are available free to both instructors and students courtesy of the Visible Software Co. These interpreters allow students to run programs on their home computers without purchasing a Pascal compiler. Interpreters are available in both a Mac and PC version and may be obtained via anonymous FTP only.

---

## FTP INSTRUCTIONS

Some of the above listed supplemental materials for this book are only available via anonymous FTP from bc.aw.com in the subdirectory, "bc/savitch/pascal." To retrieve Supplements available via anonymous FTP, FTP to bc.aw.com as follows

```
ftp bc.aw.com
```

and then log in as "anonymous". Use your e-mail address as your password. Once logged in, change to the directory for this book by typing:

```
cd bc/savitch/pascal
```

Before retrieving supplements, look at the README file to see if changes have been made since this book went to press.

```
get README
```

will retrieve this file. Quit ftp to log off and read the file. (Although the README file can be read online, it is courteous not to tie up the login port for reading.) Then log back on when you are ready to download. You can also get a listing of available file names using either the convention UNIX "ls" command, or the DOS "dir" command.

Using anonymous FTP and then de-archiving files can get complicated. Instructions vary as to whether you are downloading Macintosh, DOS, or UNIX files. If you are new to using anonymous FTP, it is best to consult your instructor, your local Internet expert, or UNIX guru.

---

## Acknowledgments

I have received much help and encouragement from many individuals and groups while preparing this series of Pascal books. Much of the original edition was written while I was visiting at the University of Washington (Seattle) Computer Science Department. Later editions were worked on while I was visiting the Computer Science Department at the University of Cincinnati. Much of the work on this edition was done while I was visiting the Computer Science Department at the University of Colorado at Boulder. The remainder of the work on these books was done in the Computer Science and Engineering Department at the University of California, San Diego (UCSD). I am grateful to all of these institutions for providing facilities and a conducive environment for writing these books.

Special thanks go to Michael Main of the University of Colorado at Boulder who provided me with a great deal of assistance on this revision. His ideas, experience, and enthusiasm are clearly reflected in this revision.

A large number of reviewers have contributed critiques and suggestions for this or previous Pascal books in this series. Their suggestions and comments have helped tremendously in shaping this latest edition. I gratefully acknowledge their invaluable help. In alphabetical order these individuals are Guy Almes, Stephen Andrilli, Bill Appelbe, Owen Astrachan, R.W. Barton, Philip Beckman, Beverly Bilshausen, Andrew Black, Jim Bunch, Baldwin van der Byl, Phillip Carrigan, Scott Cormode, Christine

---

Coulter, Mohammad B. Dadfar, Molly E. Daniel, Lieutenant Commander Paul Desilets, Mike Denisevich, John Donald, H. Edward Donley, Chris Dovolis, Patrick Dymond, Klaus Eldridge, Eileen Entin, Roger Ehrich, David R. Falconer, Allyn Fratkin, Capt. C.C. Gardner, Jim Gips, Chia Yung Han, Paul Hanna, Robert M. Holloway, Bill Hotard, Dale W. Isner, Brian Johnson, Richard Kaufmann, Ken Kellum, Alean Kirnak, Robert M. Knodel, J. Mailen Kootsey, Gregg Kornfeld, William Kraynek, Karl Krummel, Ronald L. Lancaster, N. Lehmkul, E. M. Lether, K. W. Loach, Moira Mahony, Michael Main, Andrea Martin, Keith Muller, Rayno Niemi, James M. Ott, G. Ozsoyogui, Jerome Paul, J. F. Paris, Jo Ellen Perry, James Payne, Gary Phillips, Collette Pirie, Howard Pyron, Vijay Rao, Colonel W.A. Richardson, Ned Rosen, Robert Rother, Arden Ruttan, Gary Sackett, Joe Sandmeyer, Robert Streett, George Stockman, Lieutenant Colonel Scott C. Teel, Rena Tobias, Martin Tompa, Dennis Volper, Joseph Waters, Larry Weber, Gregory Wetzels, Anne Wilson, Chin Wu, and Guy Zimmerman.

I extend a special thanks to the many students in my programming classes who tested and helped correct preliminary versions of all editions in this series. I also thank all the individuals at Benjamin/Cummings who organized the reviewing and production of this book. In particular, Colleen Dunn, Teri Holden, Vivian McDougal, Melissa Standen, Mary Tudor, and especially my editor, Carter Shanklin, contributed much to the finished product.

W.S.

---

# BRIEF CONTENTS

Chapter One	<b>Introduction to Computers and Problem Solving</b>	1
Chapter Two	<b>Introduction to Problem Solving with Pascal</b>	21
Chapter Three	<b>More Pascal and Programming Techniques</b>	61
Chapter Four	<b>Designing Procedures for Subtasks</b>	107
Chapter Five	<b>Procedures for Modular Design</b>	145
Chapter Six	<b>Designing Programs that Make Choices</b>	181
Chapter Seven	<b>Design and Implementation of Loops</b>	213
Chapter Eight	<b>Functions and Abstract Data Types</b>	255
Chapter Nine	<b>One-Dimensional Arrays</b>	317
Chapter Ten	<b>Complex Array Structures</b>	353
Chapter Eleven	<b>Records and Other Data Structures</b>	401
Chapter Twelve	<b>Program Design Methodology</b>	453
Chapter Thirteen	<b>Text Files and Secondary Storage</b>	483
Chapter Fourteen	<b>Problem Solving Using Recursion</b>	515
Chapter Fifteen	<b>Solving Numeric Problems</b>	559
Chapter Sixteen	<b>More File Types</b>	585
Chapter Seventeen	<b>Dynamic Data Structures</b>	615
Appendix One	<b>The goto Statement</b>	A-1
Appendix Two	<b>Syntax Diagrams for Standard Pascal</b>	A-4
Appendix Three	<b>Using Files with Dr. Pascal</b>	A-19
Appendix Four	<b>Getting Started with UNIX</b>	A-22
Appendix Five	<b>A Quick Introduction to the Vi Editor</b>	A-23
Appendix Six	<b>Summary of Vi Commands</b>	A-28
Appendix Seven	<b>UNIX Files and Directories</b>	A-34
Appendix Eight	<b>Summary of UNIX Commands</b>	A-39
Appendix Nine	<b>I/O Redirection in UNIX</b>	A-41
Appendix Ten	<b>Character Sets</b>	A-43
Answers to Self-Test Exercises		A-45

---

Index	I-1
Description of Dr. Pascal, Free PCPix/MacPix Command-line interpreters, and Discount Purchase Coupon	
Inside of Back Cover:	
Reserved Words	
Standard Identifiers	
Precedence of Operators	
Ordinal Types	
Types that May Be Used for the Value Returned by a Function	
Predefined Functions	

---

---

# CONTENTS

## CHAPTER ONE

### **Introduction to Computers and Problem Solving** **1**

---

What a Computer Is	3
The Modern Digital Computer	3
The Notion of an Algorithm	8
Programs and Data	9
High Level Languages	10
The Pascal Language	12
Programming Environments	13
Designing Programs	14
Summary of Terms	17
Exercises	18
References for Further Reading	20

## CHAPTER TWO

### **Introduction to Problem Solving with Pascal** **21**

---

The Notion of a Program Variable	23
Stepping Through a Program	23
Assignment Statements	26
<i>Pitfall</i> —Uninitialized Variables	27
Data Types—An Introduction	28
More about Real Values	31
Type Compatibilities	32
Arithmetic Expressions	32
Simple Output	35
Input	36
Designing Input and Output	38
<i>Pitfall</i> —Input in Wrong Order	38
Names: Identifiers	38
Putting the Pieces Together	40
Introduction to Programming Style	41

---

Self-Test Exercises	43
Interactive Exercises	44
Problem Solving and Program Design	44
Top-Down Design	45
Case Study—A Guessing Game	45
Integer Division— <i>mod</i> and <i>div</i>	46
Desktop Testing	48
Case Study—Making Change	48
Exploring the Solution Space	53
Summary of Problem Solving Techniques	53
Summary of Programming Techniques	54
Summary of Pascal Constructs	54
Exercises	57

### CHAPTER THREE

## More Pascal and Programming Techniques

61

Naming Constants	63
Comments	64
Pitfall—Forgetting a Closing Comment Delimiter	65
Formatted Output	65
Example Using Named Constants and Formatted Output	67
Allowable Range for Numbers	68
More about Commenting	69
Testing and Debugging	70
Tracing	73
Use of Assertions in Testing (Optional)	74
Self-Test Exercises	75
Interactive Exercises	75
Syntax Diagrams	75
Simple Branching— <i>if-then-else</i>	77
Pitfall—The Equality Operator	80
Optional <i>else</i>	81
Pitfall—Extra Semicolons	83
Compound Statements	83
Iterative Enhancement	84
Case Study—Payroll Calculation	84
Simple Looping— <i>while</i> Statements	88
<i>while</i> Loop Example—Charge Card Balance	90
Pitfall—Infinite Loops	92
Standard Functions	93
Using Known Algorithms	94
Case Study—Solving Quadratic Equations	95
Defensive Programming	97

---

More about Indenting and Commenting	98
Summary of Problem Solving Techniques	98
Summary of Programming Techniques	98
Summary of Pascal Constructs	99
Exercises	101
References for Further Reading	105

## CHAPTER FOUR

### Designing Procedures for Subtasks 107

Simple Pascal Procedures	109
Variable Parameters	109
Parameter Lists	111
Implementation of Variable Parameters (Optional)	114
Procedures Calling Procedures	116
Procedural Abstraction	120
Self-Test Exercises	121
Value Parameters	123
What Kind of Parameter to Use	125
Mixed Parameter Lists	126
<i>Case Study</i> —Supermarket Pricing	128
<i>Case Study</i> —Change Program with Procedures	130
<i>Pitfall</i> —Incorrectly Ordered Parameter Lists	134
Generalizing Procedures	135
Choosing Parameter Names	136
Summary of Problem Solving and Programming Techniques	138
Summary of Pascal Constructs	138
Exercises	140

## CHAPTER FIVE

### Procedures for Modular Design 145

Local Variables	147
<i>Case Study</i> —Grade Warnings	149
Other Local Identifiers	153
<i>Pitfall</i> —Use of Global Variables	154
Self-Test Exercises	156
Implementation of Value Parameters	157
<i>Pitfall</i> —Inadvertent Local Variables	159
Scope of an Identifier	159
<i>Case Study</i> —Automobile Bargaining	164
Testing Procedures	167
Top-Down and Bottom-Up Strategies	167
Preconditions and Postconditions	173
<i>Case Study</i> —Calculating Leap Years	174



Summary of Problem Solving and Programming Techniques	176
Summary of Pascal Constructs	176
Exercises	177

**CHAPTER SIX****Designing Programs that Make Choices** **181**

Nested Statements	183
Nesting <i>if-then</i> and <i>if-then-else</i> Statements	184
Complex Boolean Expressions	184
George Boole (Optional)	187
Evaluating Boolean Expressions	187
<i>Pitfall</i> —Undefined Boolean Expressions	188
Self-Test Exercises	189
Programming with Boolean Variables	190
<i>Pitfall</i> —Omitting Parentheses in Boolean Expressions	191
Boolean Input and Output	193
<i>Case Study</i> —Designing Output	193
Boolean Constants and Debugging Switches	195
The <i>case</i> Statement	198
<i>Pitfall</i> —Case Expression Value Not on Any Label List	201
The Empty Statement	201
Programming Multiple Alternatives	202
<i>Case Study</i> —State Income Tax	203
Simple Use of Sets (Optional)	204
Summary of Problem Solving and Programming Techniques	206
Summary of Pascal Constructs	206
Exercises	208

**CHAPTER SEVEN****Design and Implementation of Loops** **213**

Basic Loop Considerations	215
The While Statement	215
<i>Case Study</i> —Summing a List of Numbers	216
<i>Pitfall</i> —Uninitialized Variables	218
<i>Pitfall Review</i> —Unintended Infinite Loops	219
Self-Test Exercises	220
The Repeat Statement	221
Comparison of the While and Repeat Loops	222
<i>Case Study</i> —Testing a Procedure	222
Self-Test Exercises	224
Designing Robust Programs	224
Invariant Assertions and Variant Expressions (Optional)	225
For Statements	229