

Introduction to Structured Programming Using

Turbo Pascal[®]

Version 5.0 on the IBM PC

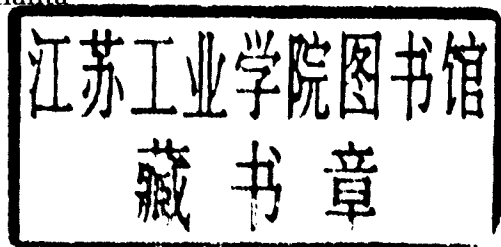


Kenneth J. Morgan

INTRODUCTION TO STRUCTURED PROGRAMMING USING TURBO PASCAL® VERSION 5.0 ON THE IBM PC

KENNETH J. MORGAN

DeVry Institute of Technology—Atlanta



Merrill Publishing Company
A Bell & Howell Information Company
Columbus Toronto London Melbourne

Cover: Computer-generated art, "Depth of Feel," by Jim Dixon, Pacific Data Images, 1986.

Published by Merrill Publishing Company
A Bell & Howell Information Company
Columbus, Ohio 43216

This book was set in Melior.

Administrative Editor: Vern R. Anthony
Production Editor: Mary Harlan
Art Coordinator: Pete Robison
Cover Designer: Cathy Watterson
Text Designer: Cynthia Brunk

Turbo Pascal is a registered trademark of Borland International, Inc.

The quotation on pp. 4–5 is excerpted from "Sage of Software," by Steve Olson.
Reprinted by permission from the Jan./Feb. issue of *Science* '84. Copyright © 1984 by
the American Association for the Advancement of Science.

Copyright © 1990 by Merrill Publishing Company. All rights reserved. No part of this
book may be reproduced in any form, electronic or mechanical, including photocopy,
recording, or any information storage and retrieval system, without permission in
writing from the publisher. "Merrill Publishing Company" and "Merrill" are
registered trademarks of Merrill Publishing Company.

Library of Congress Catalog Card Number: 89-61699
International Standard Book Number: 0-675-20770-3
Printed in the United States of America

1 2 3 4 5 6 7 8 9—94 93 92 91 90

MERRILL PUBLISHING COMPANY AND THE AUTHOR MAKE NO WARRANTY OR REPRESENTATION THAT ANY OF THE SOFTWARE PACKAGES REFERRED TO IN THE TEXT WILL FUNCTION PROPERLY IN EVERY HARDWARE/SOFTWARE ENVIRONMENT.

MERRILL PUBLISHING COMPANY AND THE AUTHOR MAKE NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO ANY OF THE SOFTWARE OR DOCUMENTATION REFERRED TO HEREIN, THEIR QUALITY, PERFORMANCE, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.

IN NO EVENT WILL MERRILL PUBLISHING COMPANY OR THE AUTHOR BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE ANY OF THE SOFTWARE OR DOCUMENTATION REFERRED TO HEREIN.

MERRILL SERIES IN COMPUTER AND INFORMATION SYSTEMS

- | | |
|-------------------------------------|---|
| BARKER | <i>Developing Business Expert Systems with LEVEL5, 20951-X</i> |
| BROQUARD/WESTLEY | <i>Fundamentals of Assembler Language Programming for the IBM PC and XT, 21058-5</i> |
| CAFOLLA/KAUFFMAN | <i>Turbo Prolog: Step by Step, 20816-5</i> |
| CHANDLER/LIANG | <i>Developing Expert Systems for Business Applications, 21102-6</i> |
| CHIRLIAN | <i>Programming in C + +, 21007-0</i> <i>Turbo Prolog: An Introduction, 20846-7</i> <i>UNIX for the IBM PC: An Introduction, 20785-1</i> |
| DENOIA | <i>Data Communication: Fundamentals and Applications, 20368-6</i> |
| GEE | <i>A Programmer's Guide to RPG II and RPG III, 20908-0</i> |
| HOBART/OCTERNAUD/ SYTSMA | <i>Hands-On Computing Using WordPerfect 5.0, Lotus 1-2-3, and dBase IV, 21110-7</i> |
| HOUSTON | <i>Looking into C, 20845-9</i> |
| INGALSBE | <i>Using Computers and Applications Software Featuring VP-Planner, dBase III/III Plus, and WordPerfect, 21097-6</i> <i>Using Computers and Applications Software Featuring Lotus 1-2-3, dBase III/III Plus, and WordPerfect, 21179-4</i> |

| | |
|-----------------------------|---|
| INGALSBE | <i>Business Applications Software for the IBM PC Alternate Edition with VP-Planner, dBase III/III Plus, and WordPerfect, 21000-3</i> <i>Business Applications Software for the IBM PC Alternate Edition with Lotus 1-2-3, dBase III/III Plus, and WordPerfect, 21042-9</i> |
| LETRAUNIK | <i>MVS/XA JCL: A Practical Approach, 20916-1</i> |
| LIPNER/KALMAN | <i>Computer Law: Cases and Materials, 21104-2</i> |
| MELLARD | <i>Introduction to Business Programming Using Pascal, 20547-6</i> |
| MORGAN | <i>Introduction to Structured Programming Using Turbo Pascal Version 5.0 on the IBM PC, 20770-3</i> |
| MORIBER | <i>Structured BASIC Programming, Second Edition, 20715-0</i> |
| REYNOLDS/REICKS | <i>Introduction to Business Telecommunications, Second Edition, 20815-7</i> |
| ROSEN | <i>Office Automation and Information Systems, 20557-3</i> |
| SCHATT | <i>Microcomputers in Business and Society, 20862-9</i> |
| SPRANKLE | <i>Problem Solving and Programming Concepts, 20867-X</i> |
| STARK | <i>The Complete Textbook of Lotus 1-2-3, 21103-4</i> |
| SZYMANSKI | <i>Computers and Information Systems in Business, 20905-6</i> |
| SZYMANSKI/SZYMANSKI/ | <i>Computers and Application Software, 20904-8</i> |
| MORRIS/PULSCHEN | <i>Introduction to Computers and Information Systems, 20768-1</i> |

Preface

This textbook was written with both students and instructors in mind. It attempts to provide all the material needed for a course having the following objectives:

1. To teach the student how to program, that is, to teach the theory and practice of structured programming and the standard techniques generally employed in programming
2. To teach the student how to use the IBM/PC
3. To teach the student how to use the PC-DOS operating system
4. To teach the student the syntax of the Pascal language
5. To teach the student how to use the Turbo Pascal system for developing programs in Pascal

To help meet all five goals in one course—an ambitious undertaking—this text contains a systematic and comprehensive explanation of structured programming and a thorough discussion of PC-DOS and the Turbo Pascal integrated development environment (IDE), as well as a complete presentation of Pascal syntax.

Organization

To obtain the maximum benefit from this textbook, it is important to understand how it is organized and to appreciate the philosophy behind that organization. A brief summary is given here; more information is provided in the instructor's manual.

The book has two parts: Part I is a thorough introduction to programming in general and structured programming in particular; Part II presents the syntax of Pascal. It is critically important to understand how Parts I and II are related to each other:

*Parts I and II are not intended to be taught
in succession, but rather simultaneously.*

Part I is organized around the phases of program development. The philosophy of structured programming is developed as each phase is explained. This part of the book emphasizes primarily the “how” of programming—something essentially language-independent.

Despite the distinct emphases of the two parts, the philosophy and programming techniques discussed in Part I are illustrated with a specific language to enhance the book's usefulness. Thus, Part I includes example Pascal code with a few brief comments on syntax. The instructor can amplify these comments or assign the appropriate sections of Part II for more details. This organization not only gives the student a clear, systematic development of the principles of structured programming, it also allows the instructor a significant degree of freedom to introduce topics of syntax and assign programming projects as he desires. Therefore, the following recommendation is made:

Part I should be taught sequentially; its twelve chapters represent the entire course.

Part II is organized around the main features of the Pascal language. It begins with two rather detailed chapters, one discussing PC-DOS and the other the Turbo Pascal IDE. The remainder of Part II is a comprehensive presentation of Pascal syntax written in the form of an expanded manual. This part is organized around Pascal syntax in much the same way as the *Pascal User Manual and Report* by Jensen and Wirth. Using this textbook, therefore, will help the student gain facility in using manuals—a significant skill in data processing. Part II is an “expanded” manual, however, in that detailed explanation is provided, along with many examples.

Major Features

Style and format.

This book is not intended for the shelf of the local computer store. It is not a self-study guide; it is a textbook. As such, it demands effort on the part of the student, and it assumes an instructor.

While effort has been made to achieve clarity of exposition, students are encouraged to use dictionaries for the nontechnical words that may be new to them. I have long decried the slow erosion of proficiency in language use and grammar. To write a text that uses no words outside the present vocabulary of the students who use it and that contains no complex or compound sentences is a disservice to them. A student's task is to learn new things. A true education is one in which the student's ability to use language grows with progress in his chosen field.

Required background and mathematical level.

A general knowledge of computer terminology is helpful, but no specific programming experience is assumed. The level of the text is suitable for a beginning programming class in college. The mathematical background assumes only algebra and a rudimentary knowledge of trigonometry.

This book takes a generic approach to applications; it is neither engineering oriented nor business oriented. None of the relatively simple examples and programming assignments represent technical aspects of any field. Rather, they include elementary topics useful in both scientific and business programming.

Avoidance of “case studies.”

Because extended “case studies” of long programs do not represent the best way to illustrate either programming principles or points of syntax, the illustrations in this

book are short and simple. The techniques represented can, however, be incorporated into longer programming assignments that the students themselves write.

For instructors who like to present case studies in their lectures, the instructor's manual includes solutions to all the programming assignments in Appendix B. Some of the more lengthy programs can be reproduced and distributed to the students.

Exercises, problems, and programming assignments.

At the end of each section of each chapter, there is a set of simple review questions. These are not "thought questions" or research projects. They simply break the monotony of reading and give students the opportunity to check their comprehension of the previous section before proceeding. They direct students to the important points in that section.

Most chapters also conclude with a set of problems. In Part II, these include many short programs that require the student to find an error or determine the output.

The end-of-chapter problems also include many programming assignments in which students apply what they have learned in the chapter. The resulting programs vary in length. Although detailed specifications and formatting requirements are not given for these assignments, Appendix B does contain a number of thoroughly described programming assignments from which the instructor can choose. These programs also vary in length. To aid the instructor in choosing appropriate assignments, each project begins with a statement of purpose that summarizes the syntactical requirements. Detailed specifications for each program are then provided, often with extended discussions of how the student might write the program.

Additional features.

- Each chapter begins with an outline of the chapter sections.
- Many chapters, especially in Part I, conclude with a list of suggested outside readings. These readings may include books that give a more detailed discussion of particular topics covered in the chapter, or they may include important articles in the history of structured programming. Instructors are encouraged to assign these readings.
- Proper technical terminology is emphasized throughout. In Part II, dealing with Pascal syntax, standard Pascal terminology is defined and regularly used in describing the language. The more important definitions are set off for easy reference. They have not simply been copied from the ANSI or OSI standards but have been carefully tailored to the overall development presented in this book.

Contents

Part I deals with programming in general and specific programming procedures and techniques. It is the heart of this book, and it explains the philosophy now known as *structured programming*.

Chapter 1 contains a general introduction to structured programming: its history, rationale, definition, and advantages. Some rather interesting stories and accounts are related in this chapter. It should build interest in the subject. Chapter 2 introduces an important feature of this book—six specific steps that guide students through the entire process of structured programming. Chapters 3 through 12 explain and amplify each of the six steps in this "programming sequence."

Chapter 3 is devoted to the concept of *problem solving* in general and *top-down analysis* in particular without any direct use of programs. The hierarchy chart is also introduced as a means of summarizing the result of a top-down analysis and as a guide to the organization of a program implementing the solution.

Chapter 4 explains modular structure for programs and how to implement it using Pascal procedures. Both flowcharts and pseudocode are discussed in chapter 5; the instructor can choose whichever technique he prefers. Chapter 6, in turn, introduces the fundamental control structures used in writing structured code, while chapter 7 discusses how to “build” a structured module using the control structures.

Chapters 8 through 10 explain some of the standard programming “tricks of the trade,” such as how to use counters, accumulators, switches, sentinel values, nesting, and complex decisions; how to write menu-driven programs; how to check input data; how to use headers and perform pagination; and how to sort and search arrays. Chapter 11 introduces students to file processing in Pascal. Finally, chapter 12 discusses how to debug, test, and document a program.

In Part II, Chapter 13 contains a thorough introduction to IBM PC-DOS, while chapter 14 explains the Turbo Pascal IDE. Chapter 15 provides a general introduction to the Pascal language, and chapter 16 discusses operators, functions, expressions, and data types. Chapter 17 explains simple monitor screen I/O, and chapter 18 presents the syntax for all of the statements available in Pascal. Chapters 19 and 20 explain the syntax of functions and procedures. Chapter 21 covers user-defined scalar and subrange data types, while chapters 22 through 26 discuss Pascal’s structured data types. Chapter 27 consists of a brief introduction to pointers and dynamic storage allocation, together with a discussion of linked lists.

Appendix A contains four preliminary laboratory exercises that can be used during the early weeks of class. They contain step-by-step directions that introduce students to the operation of the IBM/PC, PC-DOS, and the Turbo Pascal IDE, with emphasis on using the editor. The last exercise includes a simple but complete Pascal program to type and run. Appendix B contains detailed specifications for 18 programming assignments. Appendix C contains an introduction to every feature of the Turbo Pascal integrated debugger.

Acknowledgments

I wish to thank my wife, Carol, for invaluable help in proofreading the manuscript, and my colleague, Linda Isabel, for her faithful classroom use of this book during its entire development. I also would like to thank my administrative editor at Merrill Publishing, Vern Anthony, for his patience and understanding through a very long development process. I also thank my production editor, Mary Harlan, who was able to keep track of the many changes that occurred during development and who taught me the difference between *that* and *which*. Finally, I wish to express my appreciation to the following professors who reviewed the manuscript and made many helpful comments and suggestions: Donald Cartlidge, New Mexico State University; Robert Gann, Hartwick College; Reino Hakala, Governors State University; Merle Parmer, DeVry Institute of Technology–Phoenix; Yoseph Qarmaza, DeVry Institute of Technology–Lombard; Nancy Quesenberry, Allan Hancock College; Ira Scheer, DeVry Institute of Technology–Columbus; Bruce Sisko, Belleville Area College; and Sharon Szabo, Schoolcraft College.

Contents

PART ONE COMPUTER PROGRAMMING

| | | |
|----------|---|----------|
| 1 | Structured Programming | 3 |
| 1.1 | Structured Programming: History and Rationale | 3 |
| 1.2 | Definition of Structured Programming | 6 |
| 1.3 | Advantages of Structured Programming | 9 |
| 1.4 | Structured Programming and Programming Languages | 10 |
| 1.5 | Additional Topics Related to Structured Programming | 11 |

| | | |
|----------|----------------------------------|-----------|
| 2 | The Programming Sequence | 13 |
| 2.1 | Problem Solving and Computers | 13 |
| 2.2 | Programming and Systems Analysis | 15 |
| 2.3 | The Programming Sequence | 15 |

| | | |
|----------|---|-----------|
| 3 | Top-down Analysis and Hierarchy Charts | 21 |
| 3.1 | Problem Solving and Algorithm Design | 21 |
| 3.2 | Definition of Top-down Analysis | 22 |
| 3.3 | Four Basic Steps to Top-down Analysis | 24 |
| 3.4 | Top-down Analysis and Outlining | 26 |
| 3.5 | The Hierarchy Chart | 27 |
| 3.6 | Examples of Top-down Analysis | 29 |

| | | |
|----------|-------------------------------|-----------|
| 4 | Modular Programs | 36 |
| 4.1 | Block-structured Languages | 37 |
| 4.2 | Structure of a Pascal Program | 38 |

- 4.3 Important Programming Terminology 39
- 4.4 Definition of Modular Programming 44
- 4.5 The Relationship between Modules and Pascal Subroutines 45
- 4.6 The Two Methods of Branching Contrasted 46
- 4.7 Top-down Coding 49
- 4.8 Design Principles for Modular Programming 50
- 4.9 Reading Hierarchy Charts 54

5 Flowcharts and Pseudocode 63

- 5.1 Introduction to Flowcharts 63
- 5.2 Flowchart Symbols 65
- 5.3 Rules for Flowcharting 67
- 5.4 Formal Example: EVEN OR ODD Program 70
- 5.5 Modules for EVEN OR ODD Program 74
- 5.6 A Candid Evaluation of Flowcharts 76
- 5.7 Introduction to Pseudocode 78

6 Control Structures 86

- 6.1 Recapitulation and Introduction 86
- 6.2 The Building Blocks of Modules 87
- 6.3 Translating a Flowchart and Pseudocode into Pascal 89
- 6.4 Definition of a Control Structure 92
- 6.5 Control Structure Names 95
- 6.6 Characteristics of Control Structures 96
- 6.7 Nonrepetitive Control Structures 97
- 6.8 Repetitive Control Structures 107

7 Structured Code: How to Build a Module 124

- 7.1 Definition of Structured Code 124
- 7.2 Building a Module: A First Look at Nesting 126
- 7.3 Examples of Building Structured Modules 127
- 7.4 When a Structured Control Statement Is Not a Control Structure 133
- 7.5 Simulating Other Control Structures 138
- 7.6 Thinking in Terms of Control Structures 142
- 7.7 GOTO: To Use or Not to Use—That Is the Question 144
- 7.8 The Art and Science of Computer Programming 145

8 Elementary Programming Techniques 155

- 8.1 Monitor-Screen Prompts 156
- 8.2 Repeating the Control Section 159
- 8.3 Pausing 161
- 8.4 Initialization 162
- 8.5 The Use of Counters 163
- 8.6 The Use of Accumulators 166

- 8.7 The Use of Switches and Flags 167
- 8.8 The Use of Sentinel Values 170

| | | |
|----------|--|------------|
| 9 | Intermediate Programming Techniques | 178 |
| 9.1 | Strings and String Operations | 178 |
| 9.2 | Arrays and Array Operations | 185 |
| 9.3 | The Use of Complex Conditions | 192 |
| 9.4 | A Second Look at Nesting | 195 |
| 9.5 | Complex Decisions | 202 |
| 9.6 | Menu-Driven Programs | 208 |
| 9.7 | Rounding | 212 |
| 9.8 | Checking Input Data | 216 |

| | | |
|-----------|--|------------|
| 10 | Advanced Programming Techniques | 227 |
| 10.1 | Headers and Pagination | 227 |
| 10.2 | Sorting | 235 |
| 10.3 | Searching | 239 |
| 10.4 | Scaling | 244 |
| 10.5 | Recursion | 250 |

| | | |
|-----------|--|------------|
| 11 | Records and Files | 258 |
| 11.1 | Definition of a File | 259 |
| 11.2 | Definition of a Record | 260 |
| 11.3 | Relationship between Records and Arrays | 261 |
| 11.4 | Accessing Fields | 263 |
| 11.5 | I/O Operations with Records | 265 |
| 11.6 | Relationship between Files and Arrays | 266 |
| 11.7 | Computer Files and Office Files | 267 |
| 11.8 | Sequential Access and Random Access Files | 269 |
| 11.9 | General File Operations | 269 |
| 11.10 | Advanced File Operations | 279 |
| 11.11 | Random Access for Binary Files: The SEEK Procedure | 279 |
| 11.12 | Introduction to Pascal Text Files | 281 |

| | | |
|-----------|--|------------|
| 12 | Debugging, Testing, and Documentation | 294 |
| 12.1 | Distinction between Debugging and Testing | 295 |
| 12.2 | Types of Errors | 296 |
| 12.3 | Debugging Strategies | 298 |
| 12.4 | Common Causes of Programming Errors | 299 |
| 12.5 | Debugging Tricks and Techniques | 304 |
| 12.6 | Introduction to Program Testing | 307 |
| 12.7 | Constructing Test Data | 307 |
| 12.8 | Testing Strategies | 310 |
| 12.9 | Survey of Documentation | 316 |

- 12.10 The Hierarchy Chart 318
- 12.11 Well-formatted Source Listings 323
- 12.12 Program Comments 326
- 12.13 Minimum Guidelines and Requirements 331

PART TWO

SUMMARY OF PASCAL

| | | |
|-----------|--|------------|
| 13 | Introduction to IBM PC-DOS | 341 |
| 13.1 | What Is IBM PC-DOS? 342 | |
| 13.2 | What Are Diskette Files? 342 | |
| 13.3 | Three Types of Diskettes 343 | |
| 13.4 | Loading DOS 345 | |
| 13.5 | Disk Drive Names 346 | |
| 13.6 | DOS File Names 346 | |
| 13.7 | Types of Diskette Files 347 | |
| 13.8 | Diskette Organization 352 | |
| 13.9 | Syntactical Notation for DOS Commands 354 | |
| 13.10 | Formatting Diskettes 354 | |
| 13.11 | Commonly Used Internal DOS Commands 356 | |
| 13.12 | The DISKCOPY Command 360 | |
| 13.13 | Batch Files 361 | |
| 14 | The Turbo Pascal System | 363 |
| 14.1 | An Overview of the Turbo Pascal System 364 | |
| 14.2 | Installing Turbo Pascal: Working Diskettes 367 | |
| 14.3 | Loading Turbo Pascal 368 | |
| 14.4 | The Main Turbo Pascal Screen 368 | |
| 14.5 | The Turbo Pascal System of Menus 372 | |
| 14.6 | The Turbo Pascal Editor: An Introduction 375 | |
| 14.7 | The Turbo Pascal Editor: Common Commands 378 | |
| 14.8 | Typing, Compiling, and Running Programs 382 | |
| 14.9 | Turbo Pascal Error Messages 384 | |
| 14.10 | Program Units 387 | |
| 14.11 | Additional Turbo Pascal Menu Items 391 | |
| 14.12 | Compiler Directives 393 | |
| 15 | An Overview of the Pascal Language | 397 |
| 15.1 | Pascal and Turbo Pascal 397 | |
| 15.2 | Standard Language Elements 398 | |
| 15.3 | User-defined Language Elements 402 | |
| 15.4 | General Characteristics of the Pascal Language 407 | |
| 15.5 | The Structure of a Pascal Program 408 | |
| 15.6 | The Declaration Block 410 | |
| 15.7 | The Control Section 416 | |

- 15.8 Guidelines for Indentation 417
- 15.9 The Pascal Program Mainline 417

| | | |
|-----------|---|------------|
| 16 | Data Types, Operators, and Expressions | 421 |
| 16.1 | Data Types, Operators, Functions, and Expressions | 421 |
| 16.2 | Classification of Data Types | 425 |
| 16.3 | Standard Scalar Data Types | 428 |
| 16.4 | Ordinal Data Types | 433 |
| 16.5 | Assignment Compatibility | 434 |
| 16.6 | Classification of Pascal Operators | 437 |
| 16.7 | Classification of Pascal Expressions | 441 |
| 16.8 | The Rules of Operator Precedence | 441 |

| | | |
|-----------|---|------------|
| 17 | Simple Input/Output | 449 |
| 17.1 | Introduction to the Files INPUT and OUTPUT | 450 |
| 17.2 | General Characteristics of the READ/READLN and WRITE/WRITELN Statements | 451 |
| 17.3 | The WRITE/WRITELN Statements | 452 |
| 17.4 | The READ/READLN Statements | 457 |
| 17.5 | The READKEY Statement | 464 |

| | | |
|-----------|---|------------|
| 18 | Pascal Statements | 472 |
| 18.1 | Classification of Pascal Statements | 472 |
| 18.2 | The Assignment Statement | 474 |
| 18.3 | The Procedure and GOTO Statements | 475 |
| 18.4 | The Null Statement | 477 |
| 18.5 | The Compound Statement | 478 |
| 18.6 | The IF Statement | 479 |
| 18.7 | The CASE Statement | 482 |
| 18.8 | The FOR Statement | 486 |
| 18.9 | The WHILE Statement | 489 |
| 18.10 | The REPEAT Statement | 490 |
| 18.11 | Comparison of the Three Loop Statements | 492 |

| | | |
|-----------|--|------------|
| 19 | Predefined Procedures and Functions | 498 |
| 19.1 | Predefined and User-defined Subroutines | 498 |
| 19.2 | Mathematical Functions and Programming Functions | 499 |
| 19.3 | Calling Procedures and Functions | 500 |
| 19.4 | The Use of Predefined Functions | 502 |
| 19.5 | Calling Predefined Functions: Syntax Summary | 503 |
| 19.6 | The Distinction between a Procedure and a Function | 504 |
| 19.7 | The Use of Predefined Procedures | 505 |
| 19.8 | Calling Predefined Procedures: Syntax Summary | 507 |

| | | |
|-----------|---|------------|
| 20 | User-defined Procedures and Functions | 510 |
| 20.1 | The Structure of a Pascal Subroutine | 510 |
| 20.2 | An Example Function Subroutine | 513 |
| 20.3 | An Example Procedure Subroutine | 514 |
| 20.4 | Passing Parameters to and from Procedures | 516 |
| 20.5 | Procedure Declaration and Calling: Syntax Summary | 520 |
| 20.6 | Function Declaration and Calling: Syntax Summary | 522 |
| 20.7 | The Rules of Scope | 524 |
| 20.8 | The FORWARD Directive | 529 |
| 20.9 | Guidelines for the Use of Pascal Variables | 531 |
| 21 | Enumerated and Subrange Scalar Data Types | 540 |
| 21.1 | Enumeration: The Basis of a Scalar Data Type | 540 |
| 21.2 | Enumerated Scalar Data Types | 541 |
| 21.3 | Enumerated Data Type Declaration: Syntax Summary | 546 |
| 21.4 | I/O with Enumerated Data Types | 547 |
| 21.5 | Subrange Scalar Data Types | 548 |
| 21.6 | Subrange Data Type Declaration: Syntax Summary | 549 |
| 21.7 | Index Range Checking | 550 |
| 22 | Array Data Types | 554 |
| 22.1 | Definition of an Array | 554 |
| 22.2 | Array Declaration: Syntax Summary | 555 |
| 22.3 | Assignment Compatibility | 557 |
| 22.4 | Arrays of Characters | 560 |
| 22.5 | Multidimensional Arrays | 563 |
| 23 | String Data Types | 570 |
| 23.1 | Definition of a String | 570 |
| 23.2 | String Declaration: Syntax Summary | 572 |
| 23.3 | Rules on Truncation and Concatenation | 574 |
| 23.4 | Assignment Compatibility | 575 |
| 23.5 | Comparison Operations | 576 |
| 23.6 | Standard Functions and Procedures | 577 |
| 24 | Set Data Types | 587 |
| 24.1 | Definition of a Set | 587 |
| 24.2 | Set Declaration: Syntax Summary | 588 |
| 24.3 | Set Assignment | 590 |
| 24.4 | Operators Used with Sets | 592 |
| 24.5 | Set Expressions | 595 |
| 24.6 | The Empty Set, Unit Set, and Universal Set | 596 |

- 24.7 Assignment Compatibility 597
- 24.8 The Use of Sets 599

| | | |
|-----------|--|------------|
| 25 | Record Data Types | 608 |
| 25.1 | Definition of a Record | 608 |
| 25.2 | Record Declaration: Syntax Summary | 609 |
| 25.3 | Assignment Compatibility | 611 |
| 25.4 | Nested Records | 612 |
| 25.5 | The Scope of Field Identifiers | 613 |
| 25.6 | The WITH Statement | 615 |
| 25.7 | The WITH Statement: Syntax Summary | 620 |
| 25.8 | Variant Records | 621 |
| 25.9 | Variant Record Declaration: Syntax Summary | 626 |

| | | |
|-----------|--|------------|
| 26 | File Data Types | 634 |
| 26.1 | Definition of a File | 634 |
| 26.2 | Standard Pascal and Turbo Pascal | 636 |
| 26.3 | File Declaration: Syntax Summary | 637 |
| 26.4 | Assignment Compatibility | 640 |
| 26.5 | File-handling Routines | 640 |
| 26.6 | Binary Files: File-handling Routines | 646 |
| 26.7 | General Characteristics of Pascal Text Files | 646 |
| 26.8 | Text Files: File-handling Routines | 647 |
| 26.9 | DOS Devices | 648 |

| | | |
|-----------|---|------------|
| 27 | Pointer Variables and Dynamic Storage Allocation | 660 |
| 27.1 | Static Variables and Dynamic Variables | 660 |
| 27.2 | Pointer Variables | 661 |
| 27.3 | Declaring Pointer Variables and Dynamic Variables | 662 |
| 27.4 | Nodes and Records | 665 |
| 27.5 | Linked Lists | 666 |
| 27.6 | Creating a Linked List | 668 |
| 27.7 | Traversing a Linked List | 670 |

| | | |
|-------------------|--|------------|
| Appendix A | Preliminary Laboratory Exercises | 677 |
| A.1 | The Care and Handling of Diskettes and Disk Drives | 677 |
| A.2 | Important Note to the Student | 678 |
| A.3 | Exercise 1: Preparing Your Diskettes | 678 |
| A.4 | Exercise 2: Exploring DOS | 680 |
| A.5 | Exercise 3: Using the Turbo Menu and Editor | 682 |
| A.6 | Exercise 4: Typing an Example Pascal Program | 686 |