GLOBAL
EDITION

Problem Solving and Program Design in C

EIGHTH EDITION

Jeri R. Hanly • Elliot B. Koffman

ALWAYS LEARNING

PEARSON

EIGHTH EDITION

GLOBAL EDITION

PROBLEM SOLVING AND PROGRAM DESIGN

in C

Problem Solving and Program Design in C, Global Edition

Table of Contents

Cover

Title Page

Copyright Page

Dedication

Preface

Contents

Chapter 0: Computer Science as a Career Path

- Section 1: Why Computer Science May Be the Right Field for You
- Section 2: The College Experience: Computer Disciplines and Majors to Choose From
- Section 3: Career Opportunities

Chapter 1: Overview of Computers and Programming

- 1.1. Electronic Computers Then and Now
- 1.2. Computer Hardware
- 1.3. Computer Software
- 1.4. The Software Development Method
- 1.5. Applying the Software Development Method
  - Case Study: Converting Miles to Kilometers
- 1.6. Professional Ethics for Computer Programmers
- Chapter Review

Chapter 2: Overview of C

Pearson

Table of Contents

2.1. C Language Elements

2.2. Variable Declarations and Data Types

2.3. Executable Statements

2.4. General Form of a C Program

2.5. Arithmetic Expressions

Case Study: Supermarket Coin Processor

2.6. Formatting Numbers in Program Output

2.7. Interactive Mode, Batch Mode, and Data Files

2.8. Common Programming Errors

Chapter Review

Chapter 3: Top-Down Design with Functions

3.1. Building Programs from Existing Information

Case Study: Finding the Area and Circumference of a Circle

Case Study: Computing the Weight of a Batch of Flat Washers

3.2. Library Functions

3.3. Top-Down Design and Structure Charts

Case Study: Drawing Simple Diagrams

3.4. Functions Without Arguments

3.5. Functions with Input Arguments

3.6. Introduction to Computer Graphics (Optional)

3.7. Common Programming Errors

Chapter Review

Chapter 4: Selection Structures: if and switch Statements

4.1. Control Structures

4.2. Conditions

4.3. The if Statement

4.4. If Statements with Compound Statements

Table of Contents

- 4.5. Decision Steps in Algorithms
  - Case Study: Water Bill Problem
- 4.6. More Problem Solving
  - Case Study: Water Bill with Conservation Requirements
- 4.7. Nested if Statements and Multiple-Alternative Decisions
- 4.8. The switch Statement
  - C in Focus: The UNIX Connection
- 4.9. Common Programming Errors
- Chapter Review

Chapter 5: Repetition and Loop Statements

- 5.1. Repetition in Programs
- 5.2. Counting Loops and the while Statement
- 5.3. Computing a Sum or a Product in a Loop
- 5.4. The for Statement
- 5.5. Conditional Loops
- 5.6. Loop Design
- 5.7. Nested Loops
- 5.8. The do-while Statement and Flag-Controlled Loops
- 5.9. Iterative Approximations
  - Case Study: Bisection Method for Finding Roots
- 5.10. How to Debug and Test Programs
  - C in Focus: Team Programming
- 5.11. Loops in Graphics Programs (Optional)
- 5.12. Common Programming Errors
- Chapter Review

Chapter 6: Pointers and Modular Programming

- 6.1. Pointers and the Indirection Operator

Table of Contents

6.2. Functions with Output Parameters

6.3. Multiple Calls to a Function with Input/Output Parameters

6.4. Scope of Names

6.5. Formal Output Parameters as Actual Arguments

6.6. Problem Solving Illustrated

Case Study: Collecting Area for Solar-Heated House

Case Study: Arithmetic with Common Fractions

6.7. Debugging and Testing a Program System

6.8. Common Programming Errors

Chapter Review

Chapter 7: Array Pointers

7.1. Declaring and Referencing Arrays

7.2. Array Subscripts

7.3. Using for Loops for Sequential Access

7.4. Using Array Elements as Function Arguments

7.5. Array Arguments

7.6. Searching and Sorting an Array

7.7. Parallel Arrays and Enumerated Types

7.8. Multidimensional Arrays

7.9. Array Processing Illustrated

Case Study: Summary of Hospital Revenue

7.10. Graphics Programs with Arrays (Optional)

7.11. Common Programming Errors

Chapter Review

Chapter 8: Strings

8.1. String Basics

8.2. String Library Functions: Assignment and Substrings

Table of Contents

8.3. Longer Strings: Concatenation and Whole-Line Input

8.4. String Comparison

C in Focus: Defensive Programming

8.5. Arrays of Pointers

8.6. Character Operations

8.7. String-to-Number and Number-to-String Conversions

8.8. String Processing Illustrated

Case Study: Text Editor

8.9. Common Programming Errors

Chapter Review

Chapter 9: Recursion

9.1. The Nature of Recursion

9.2. Tracing a Recursive Function

9.3. Recursive Mathematical Functions

9.4. Recursive Functions with Array and String Parameters

Case Study: Finding Capital Letters in a String

Case Study: Recursive Selection Sort

9.5. Problem Solving with Recursion

Case Study: Operations on Sets

9.6. A Classic Case Study in Recursion: Towers of Hanoi

9.7. Common Programming Errors

Chapter Review

Chapter 10: Structure and Union Types

10.1. User-Defined Structure Types

10.2. Structure Type Data as Input and Output Parameters

10.3. Functions Whose Result Values Are Structured

C in Focus: Evolving Standards

Table of Contents

10.4. Problem Solving with Structure Types

Case Study: A User-Defined Type for Complex Numbers

10.5. Parallel Arrays and Arrays of Structures

Case Study: Universal Measurement Conversion

10.6. Union Types (Optional)

10.7. Common Programming Errors

Chapter Review

Chapter 11: Text and Binary File Pointers

11.1. Input/Output Files: Review and Further Study

11.2. Binary Files

11.3. Searching a Database

Case Study: Database Inquiry

11.4. Common Programming Errors

Chapter Review

Chapter 12: Programming in the Large

12.1. Using Abstraction to Manage Complexity

12.2. Personal Libraries: Header Files

12.3. Personal Libraries: Implementation Files

12.4. Storage Classes

12.5. Modifying Functions for Inclusion in a Library

12.6. Conditional Compilation

12.7. Arguments to Function main

12.8. Defining Macros with Parameters

12.9. Common Programming Errors

Chapter Review

Chapter 13: Pointers and Dynamic Data Structures

Table of Contents

13.1. Pointers

13.2. Dynamic Memory Allocation

13.3. Linked Lists

13.4. Linked List Operators

13.5. Representing a Stack with a Linked List

13.6. Representing a Queue with a Linked List

13.7. Ordered Lists

Case Study: Maintaining an Ordered List of Integers

13.8. Binary Trees

13.9. Common Programming Errors

Chapter Review

Appendix A: More About Pointers

Appendix B: ANSI C Standard Libraries

Appendix C: C Operators

Appendix D: Character Sets

Appendix E: ANSI C Reserved Words

Glossary

Index

Back Cover

Pearson