STRUCTURED AND OBJECT-ORIENTED PROBLEM SOLVING USING C++

Third Edition

Andrew C. Staugaard, Jr.
# GETTING ACQUAINTED WITH COMPUTERS, PROGRAMS, AND C++

## Chapter Objectives
- **Introduction** 2

## 1.1 The Hardware
- The Central Processing Unit (CPU) 2
- The Arithmetic and Logic Unit (ALU) 3
- The Control Unit 4
- Internal Registers 5
- Primary Memory 5
- Random-Access Memory (RAM) 6
- Read-Only Memory (ROM) 7
- Secondary Memory 8
- Input 9
- Output 10
- The Fetch/Execute Cycle 10

## 1.2 Programming and Software
- Machine Language 12
- Assembly Language 13
- Operating Systems 13
- High-Level Language 14
- Why C++? 16
- Some C++ History 17
- The Art and Science of Programming 18

## 1.3 Getting Started with C++: “hello world” 19
- The Preprocessor Section 20
- The Main Function Section 22

## Chapter Summary 23

## Questions and Problems 24

---

Preface xvii
2 PROBLEM SOLVING, ABSTRACTION, AND STEPWISE REFINEMENT

Chapter Objectives 29
Introduction 30
2.1 The Programmer's Algorithm 30
   Defining the Problem 31
   Planning the Solution 32
   Coding the Program 32
   Debugging the Program 33
   Desk-Checking the Program 34
   Compiling the Program 34
   Running the Program 34
   Using a Debugger 35
   Documentation 36
2.2 Problem Solving Using Algorithms 37
2.3 Problem Abstraction and Stepwise Refinement 39
Problem Solving in Action: Sales Tax 40
Problem Solving in Action: Pythagorean Theorem 44
Problem Solving in Action: Credit Card Interest 46
Chapter Summary 47
Questions and Problems 48

3 DATA: TYPES, CLASSES, AND OBJECTS

Chapter Objectives 51
Introduction 52
3.1 The Idea of Data Abstraction and Classes 53
3.2 The Primitive Data Types in C++ 57
   The Integer Types 57
   The Floating-Point Types 60
   The Character Type 63
   The Boolean Type 65
3.3 Constants and Variables 66
   Defining Constants 67
   Defining Variables 69
   Boolean Variables 73
3.4 Strings 74
   String Operations 77
3.5 Enumerated Data 80
   Defining Enumerated Types 81
   The Ordering of Enumerated Data 81
Problem Solving in Action: Bank Account Processing 84
Chapter Summary 90
Questions and Problems 91
Chapter Objectives 97
Introduction 97
Console- vs. Windows-Based Programming 98

4.1 Getting Stuff Out 99
Using the cout Object 100
Getting Out Fixed Numeric Information 100
Getting Out Fixed Character and String Information 101
Getting Out Variable Information 103
Formatting the Output 104
Formatting Floating-Point Output 113
Generating Currency Output 117

4.2 Getting Stuff In 118
Reading Mixed Data Types 122
Reading Single-Character Data 124
Using get() To Read Single Character Data 127
Reading String Data 128
Using getline() to Read Strings 130
A Problem when Using getline() 133

4.3 Reading and Writing Disk Files 136
File Streams 136
Classes: The Basis for C++ Files 136
Processing a Disk File 139
Using Loops to Read and Process Files 143

Problem Solving in Action: User-Friendly Programs 146
Chapter Summary 151
Questions and Problems 152

GUI 101: SIMPLE WINDOWS I/O 157

Chapter Objectives 169
Introduction 170

5.1 Arithmetic Operators 170
Increment and Decrement Operators 174

5.2 Assignment Operators 177

5.3 Some Standard Functions in C++ 178
Conversion Functions 179
Mathematical Functions 179

Problem Solving in Action: Inventory Control 181
Problem Solving in Action: Data Communications 188
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>DECISIONS, DECISIONS, DECISIONS</td>
<td>209-249</td>
</tr>
<tr>
<td></td>
<td>Chapter Objectives</td>
<td>209-210</td>
</tr>
<tr>
<td></td>
<td>Introduction</td>
<td>210-211</td>
</tr>
<tr>
<td></td>
<td>6.1 Boolean Operators</td>
<td>210-211</td>
</tr>
<tr>
<td></td>
<td>Relational Operators</td>
<td>211-214</td>
</tr>
<tr>
<td></td>
<td>Logical Operators</td>
<td>214-216</td>
</tr>
<tr>
<td></td>
<td>Problem Solving in Action: Boolean Logic</td>
<td>216-219</td>
</tr>
<tr>
<td></td>
<td>6.2 The if Statement</td>
<td>219-224</td>
</tr>
<tr>
<td></td>
<td>6.3 The if/else Statement</td>
<td>224-227</td>
</tr>
<tr>
<td></td>
<td>Comparing Strings</td>
<td>227-233</td>
</tr>
<tr>
<td></td>
<td>6.4 NESTED if's</td>
<td>233-237</td>
</tr>
<tr>
<td></td>
<td>6.5 The switch Statement</td>
<td>237-240</td>
</tr>
<tr>
<td></td>
<td>The default Option</td>
<td>240-248</td>
</tr>
<tr>
<td></td>
<td>Problem Solving in Action: Menu-Driven Programs</td>
<td>248-249</td>
</tr>
<tr>
<td></td>
<td>Chapter Summary</td>
<td>249-256</td>
</tr>
<tr>
<td></td>
<td>Questions and Problems</td>
<td>256-263</td>
</tr>
<tr>
<td>7</td>
<td>LOOPING OPERATIONS: ITERATION</td>
<td>263-293</td>
</tr>
<tr>
<td></td>
<td>Chapter Objectives</td>
<td>263-264</td>
</tr>
<tr>
<td></td>
<td>Introduction</td>
<td>264-268</td>
</tr>
<tr>
<td></td>
<td>7.1 The while Loop</td>
<td>264-269</td>
</tr>
<tr>
<td></td>
<td>Data Entry Using while</td>
<td>269-275</td>
</tr>
<tr>
<td></td>
<td>7.2 The do/while Loop</td>
<td>271-282</td>
</tr>
<tr>
<td></td>
<td>Problem Solving in Action: Loop-Controlled Menu-Driven Programs</td>
<td>275-282</td>
</tr>
<tr>
<td></td>
<td>7.3 The for Loop</td>
<td>282-291</td>
</tr>
<tr>
<td></td>
<td>Nested Loops</td>
<td>288-295</td>
</tr>
<tr>
<td></td>
<td>Down-to for Loops</td>
<td>290-293</td>
</tr>
<tr>
<td></td>
<td>7.4 The break and continue Options</td>
<td>291-293</td>
</tr>
<tr>
<td></td>
<td>The break Statement</td>
<td>291-293</td>
</tr>
<tr>
<td></td>
<td>The continue Statement</td>
<td>293-293</td>
</tr>
</tbody>
</table>
9.1 The Structure of an Array 386
   The Array Elements 388
   The Array Indices 388
9.2 Defining One-dimensional Arrays in C++ 388
   Initializing Arrays 390
9.3 Accessing Arrays 393
   Placing Elements into One-Dimensional Arrays 393
      Direct Assignment 393
      Keyboard Entry 394
      Using Loops 394
   Copying Elements from One-Dimensional Arrays 396
      Direct Assignment to a Variable 396
      Displaying Array Elements 397
      Using Loops 397
9.4 Arrays of Characters: C-Strings 399
   Reading and Writing C-Strings 401
   C-String Functions 405
9.5 Passing Arrays and Array Elements to Functions 408
   Passing Individual Array Elements to Functions 410
Problem Solving in Action: Searching an Array Using
   Sequential Search 416
Problem Solving in Action: Sorting an Array Using
   Insertion Sort 419
Problem Solving in Action: Searching an Array Using Recursion
   (Binary Search) 424
9.6 The STL Vector Class 430
Chapter Summary 433
Questions and Problems 433

10 CLASSES AND OBJECTS IN-DEPTH 439

Chapter Objectives 439
Introduction 440
10.1 Classes and Objects 441
   The Idea of Classes and Objects 441
   Classes 443
      The Abstract Level 443
      The Implementation Level 444
      The Data Members 444
      The Function Members 445
      The Class Declaration 445
   Encapsulation and Information Hiding 446
   Objects 453
      Defining Objects 453
10.2 Member Functions 454
   Constructor Functions 457
SE104: THE OBJECT-ORIENTED PARADIGM USING UML 490

11 CLASS INHERITANCE 501

Chapter Objectives 501
Introduction 502
11.1 Why Use Inheritance? 502
11.2 Declaring and Using Derived Classes 504
   BankAccount Class 504
   Checking Account Class 505
   SuperNow Checking Account Class 505
   Savings Account Class 505
   Single versus Multiple Inheritance 517
   Using ifndef: An Implementation Detail 517
11.3 Polymorphism and Dynamic Binding 519
   Polymorphism 519
   Dynamic versus Static Binding 520
Chapter Summary 522
Questions and Problems 522

12 FILE I/O 525

Chapter Objectives 525
Introduction 526
12.1 Fundamental Concepts and Ideas 526
   Classes Provide the Basis for C++ Files 527
   Creating File Streams in C++: The File Stream Definition 527
12.2 Accessing File Information 531
The File Window 531
File Operations 531
  Obtaining A Disk File Name from the User 533
  Writing, or Creating, a New File 533
  Reading and Displaying an Existing File 535
  Appending an Existing File 536
  Changing an Existing File 538
The Application Program 542

Chapter Summary 547
Questions and Problems 548

13  POINTERS 551

Chapter Objectives 551
Introduction 552
13.1 The Idea of Pointers 552
13.2 Defining Pointers and Initializing Pointer Data 554
  Static Pointers 555
  Dynamic Pointers 557
13.3 Accessing Pointer Data and Pointer Arithmetic 559
  Pointer Arithmetic 561
13.4 Arrays of Pointers ⇒ Indirection 564
13.5 Using Pointers as Function Arguments and Parameters 567
13.6 Object Pointers 571
  Defining Object Pointers 572
  Using the Pointer Operator to Send Messages to Objects 573
  Destructors 575

Chapter Summary 579
Questions and Problems 580

14  INTRODUCTION TO DATA STRUCTURES AND ADTs 585

Chapter Objectives 585
Introduction 586
14.1 The Concept of Data Abstraction Revisited 586
14.2 The Stack ADT 589
  Implementing the Stack ADT 591
    Creating a Stack Using an Array 591
    Pushing Elements onto a Stack 592
    Popping Elements from a Stack 594
    Inspecting the Top Element of a Stack 596
    Coding the Stack ADT 596
14.3 The Queue ADT 602
  Implementing the Queue ADT 604
    Creating a Queue Using a Circular Array 604
    Inserting Elements into a Queue 605
    Removing Elements from a Queue 607
    Inspecting the Front Element of a Queue 608
    Coding the Queue ADT 610
14.4 The List ADT 615
  Linked Lists 616
    Implementing the Linked List ADT 618
      Creating an Empty Linked List 619
      Inserting a Node into a Linked List 619
      Deleting Data from a Linked List 621
      Searching a Linked List 622
      Deleting a Node in a Linked List 623
      Traversing a Linked List 624
      Checking for an Empty List 625
      Coding the Linked List ADT 625
14.5 The Standard Template Library, or STL 631
  The STL stack Class 632
  The STL queue Class 635
  The STL list Class 637
  Chapter Summary 640
  Questions and Problems 641

MULTIDIMENSIONAL ARRAYS 645

Chapter Objectives 645
Introduction 646
15.1 Two-Dimensional Arrays 646
  Defining Two-Dimensional Arrays in C++ 647
  Accessing Two-Dimensional Array Elements 650
    Direct Assignment of Two-Dimensional Array Elements 650
    Reading and Writing Two-Dimensional Array Elements 651
    Using Loops to Access Two-Dimensional Arrays 651
15.2 Arrays of More than Two Dimensions 660
  Problem Solving in Action: Simultaneous Equation Solution 663
    Determinants 664
    Expansion of a Determinant 664
    An Order 2 Determinant Expansion Function 666
    Cramer's Rule 667
      Implementing Cramer's Rule in C++ 668
  Chapter Summary 673
  Questions and Problems 673
APPENDIX A: QUICK CHECK SOLUTIONS 679

APPENDIX B: ASCII CHARACTER TABLE 703

APPENDIX C: VISUAL STUDIO TUTORIAL 705

Introduction 705
   Who Should Use This Tutorial? 705
   What Is This Tutorial? 705
   Why Microsoft Visual Studio? 706
C.1 Jump Right In: A Visual Studio Overview 707
   Getting Started 707
   Your First Visual C++ Program 707
      Creating a New C++ Source File 708
      The Project Directory 708
      Type-in the Code 710
      Build the Program 711
      Exploring the Environment 712
      Run the Program 712
C.2 Dealing with Errors in Visual C++ 713
   Syntax Errors 713
   Using the Debugger 715
      Finding Logic Errors with the Debugger 717
   Using the Watch Window 721
C.3 Visual C++ Projects 722
   Creating a New Project 723
   Adding Multiple Files to a Project 723
   Adding Classes to a Project 730
   Using the Debugger with Functions and Classes 734
C.4 Visual C++ 6 Specific Comments 735
   Conclusion 736

APPENDIX D: C++ QUICK REFERENCE WITH STL 737

   Keywords 737
   Conversion Functions 737
   Math Functions (math.h) 738
   STL list Class Functions 738
   STL stack Class Functions 739
   String Class Operations 739
CONTENTS

STL queue Class Functions 740
STL vector Class Functions 740

GLOSSARY 741

INDEX 753