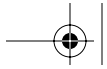


Contents

1	Software Development	1
1.1	Problem Analysis and Specification	4
1.2	Design	6
	Top-Down Design	6
	Object-Oriented Design	8
	Design in the Small	11
1.3	Coding	16
1.4	Testing, Execution, and Debugging	30
1.5	Maintenance	38
	Quick Quiz 1.5	39
	Exercises 1.5	40
	Summary	41
	Chapter Notes	41
	Programming Pointers	41
	ADT Tips	42
	Programming Problems	42
2	Introduction to Abstract Data Types	45
2.1	A First Look at ADTs and Implementations	46
2.2	C++'s Simple Data Types	46
	Integer Data	47
	Real Data	52
	Character Data	55
	Boolean Data	56
	Quick Quiz 2.2	58
	Exercises 2.2	58
2.3	Programmer-Defined Data Types	59
	Typedefs	59
	Enumerations	60
	Classes	62
	Quick Quiz 2.3	62
	Exercises 2.3	62
2.4	Pointers	63
	Declaring and Initializing Pointers	64



xii Contents

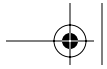
- Basic Pointer Operations 67
- Dynamic Memory Allocation—the new Operation 71
- A Note about Reference Parameters 72
- Quick Quiz 2.4 73
- Exercises 2.4 74

- Summary 75
 - Chapter Notes 75
 - Programming Pointers 76
 - ADT Tips 77
- Programming Problems 77

3 Data Structures and Abstract Data Types 81

- 3.1 Data Structures, Abstract Data Types, and Implementations 82
- 3.2 Static Arrays 85
 - One-Dimensional Static Arrays 87
 - The Subscript Operation 90
 - Arrays as Parameters 91
 - Out-of-Range Errors 92
 - Problems with Arrays 95
 - Quick Quiz 3.2 96
 - Exercises 3.2 97
- 3.3 Multidimensional Arrays (optional) 97
 - Two-Dimensional Arrays 97
 - Higher-Dimensional Arrays 99
 - Array of Array Declarations 100
 - Multidimensional Arrays as Parameters 105
 - Quick Quiz 3.3 106
 - Exercises 3.3 106
- 3.4 Dynamic Arrays 107
 - The new Operation—Dynamic Arrays 108
 - Other Uses of Pointers 121
 - Quick Quiz 3.4 122
 - Exercises 3.4 123
- 3.5 C-Style Structs (optional) 123
 - Pointers to Structs 127
 - Quick Quiz 3.5 128
- 3.6 Procedural Programming 128
 - Example of Procedural Programming 129
- Summary 135
 - Chapter Notes 135





	Programming Pointers	136
	ADT Tips	137
	Programming Problems	137
4	More about OOP and ADTs—Classes	143
4.1	Procedural vs. Object-Oriented Programming	144
	Quick Quiz 4.1	145
4.2	Classes	145
	Differences between “Traditional” (C) Structs and OOP (C++) Structs and Classes	145
	Class Declarations	146
	Quick Quiz 4.2	150
4.3	Example: A First Version of a User-Defined Time Class	150
	Why not make all class members public?	152
	Implementation of a Class	153
	Some Observations	156
4.4	Class Constructors	158
4.5	Other Class Operations	165
	Copy Operations—Initialization and Assignment	166
	Accessors and Mutators	167
	Overloading Operators	169
	Overloading Input/Output Operators	170
	Other Operations: Advance and the Relational Operators	177
	Summary and a Few Other Items	180
	Pointers to Class Objects	185
	The <code>this</code> Pointer	185
	Quick Quiz 4.5	187
	Exercises 4.5	187
	Summary	188
	Chapter Notes	188
	Programming Pointers	189
	ADT Tips	190
	Programming Problems	190
5	Standard C++ Input/Output and String Classes	193
5.1	The C++ Standard I/O Classes	194
	The <code>istream</code> Class	195
	The <code>ostream</code> Class	200
	File I/O: <code>ifstream</code> and <code>ofstream</code> Classes	204
	The I/O Class Hierarchy	206
	Quick Quiz 5.1	210





xiv Contents

- 5.2 The C++ String Types 211
 - C-Style Strings 212
 - The C++ `string` Class 216
 - String Streams 225
 - Quick Quiz 5.2 228
 - Exercises 5.2 229
- 5.3 Case Study: Text Editing 230
- 5.4 Introduction to Pattern Matching (optional) 240
- 5.5 Introduction to Data Encryption (optional) 243
 - Data Encryption Standard 245
 - Public-Key Encryption 246
- Summary 247
 - Chapter Notes 247
 - Programming Pointers 248
 - ADT Tips 249
- Programming Problems 249

6 Lists 253

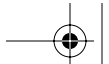
- 6.1 List as an ADT 254
 - Designing and Building a `List` Class 256
- 6.2 An Array-Based Implementation of Lists 257
 - Selecting a Storage Structure 257
 - Implementing the Operations 258
 - A `List` Class With Static Array Storage 260
- 6.3 An Array-Based Implementation of Lists with Dynamic Allocation 269
 - Dynamic Allocation in Classes—Destructors, Copy Constructors, and Assignment 275
 - A Final Note 283
 - Quick Quiz 6.3 286
 - Exercises 6.3 287
- 6.4 Introduction to Linked Lists 287
 - What Are They? 287
 - Implementing the Basic List Operations 288
 - Summary 293
 - Quick Quiz 6.4 294
 - Exercises 6.4 294
- 6.5 A Pointer-Based Implementation of Linked Lists in C++ 295
 - Node Structure 295
 - Data Members for Linked-List Implementation 297
 - Function Members for Linked-List Implementation 298
 - Exercises 6.5 301





6.6	An Array-Based Implementation of Linked Lists (optional)	303
	Node Structure	303
	Organizing the Storage Pool	306
	Exercises 6.6	308
	Summary	309
	Chapter Notes	309
	Programming Pointers	309
	ADT Tips	310
	Programming Problems	312
7	Stacks	315
7.1	Introduction to Stacks	316
	Problem 1	316
	Problem 2	316
	Problem 3	316
	Problem 4	317
7.2	Designing and Building a Stack Class—Array-Based	321
	Selecting Storage Structures	322
	Implementing the Operations	325
	The Complete Stack Class	327
	Using a Dynamic Array to Store the Stack Elements	333
	A Look Ahead	348
	Quick Quiz 7.2	349
	Exercises 7.2	350
7.3	Linked Stacks	353
	Selecting a Storage Structure	353
	Implementing the Operations	354
	The Complete Stack Class: Linked List Version	357
	Exercises 7.3	365
7.4	Use of Stacks in Function Calls	366
	Quick Quiz 7.4	370
	Exercises 7.4	370
7.5	Case Study: Postfix (RPN) Notation	371
	Evaluating Postfix Expressions	372
	Converting Infix Expressions to Postfix	374
	Quick Quiz 7.5	383
	Exercises 7.5	384
	Summary	386
	Chapter Notes	386
	Programming Pointers	386
	ADT Tips	386
	Programming Problems	386





xvi Contents



8 Queues 389

- 8.1 Introduction to Queues 389
 - Example: Drill and Practice Problems 391
- 8.2 Designing and Building a Queue Class—Array-Based 400
 - Using a Static Array to Store the Queue Elements 403
 - Using a Dynamic Array to Store the Queue Elements 408
 - Quick Quiz 8.2 409
 - Exercises 8.2 410
- 8.3 Linked Queues 412
 - A Natural Linked-List Implementation 413
 - Using a Circular Linked List 423
 - Quick Quiz 8.3 424
 - Exercises 8.3 424
- 8.4 Application of Queues: Buffers and Scheduling 426
 - Quick Quiz 8.4 429
 - Exercises 8.4 429
- 8.5 Case Study: Information Center Simulation 429
 - Problem Analysis and Specification 430
 - Building a Simulation Class 430
 - The Timer and Call Classes 441
- Summary 441
 - Chapter Notes 441
 - Programming Pointers 442
 - ADT Tips 442
- Programming Problems 442



9 ADT Implementations: Templates and Standard Containers 445

- 9.1 Introduction: The Evolution of Reusability and Genericity 446
 - From Algorithms to Algorithms 446
 - From Data to Containers 448
- 9.2 Function Genericity—Overloading and Templates 448
 - Overloading 449
 - Function Templates 451
 - Another Example: Displaying an Array 455
- 9.3 Class Genericity—Templates 457
 - What's Wrong with typedef? 457
 - Class Templates 458
 - An Alternative Version of the Stack Class Template 472
 - A Quick Peek at the Standard C++ Container Class Templates 473





- Quick Quiz 9.3 474
- Exercises 9.3 475
- 9.4 The vector Container 475
 - Declaring vector Objects 476
 - Some vector Operations 478
 - A First Look Under the Hood—Increasing the Capacity 482
 - A First Look at Iterators 485
 - Some vector Methods Involving Iterators 489
 - Wrap-up: vectors versus Arrays 490
 - Quick Quiz 9.4 491
 - Exercises 9.4 492
- 9.5 Case Study: Counting Computer Logins 494
- 9.6 Multidimensional vectors (Optional) 499
 - Two-Dimensional vector Objects 499
 - Two-Dimensional vector Operations 500
 - Exercises 9.6 503
- 9.7 Other Standard Containers—deque, stack, and queue 503
 - STL's deque Class Template 503
 - A New (But Unnecessary) Version of Our Stack Class Template 507
 - STL's stack Adapter 509
 - STL's queue Adapter 510
 - Quick Quiz 9.7 510
- 9.8 Bitsets and Valarrays (optional) 511
 - Bitsets 511
 - Valarrays 513
 - Slices, Masks, and Indirect Arrays 515
- Summary 516
 - Chapter Notes 516
 - Programming Pointers 516
 - ADT Tips 517
- Programming Problems 517

10 ADT Implementation: Recursion, Algorithm Analysis, and Standard Algorithms 521

- 10.1 Recursion 522
 - Examples of Recursive Functions 522
 - Coding Recursive Functions 524
 - Example of Inappropriate Recursion: Fibonacci Numbers 527
 - Example: Binary Search 529
 - Example: Palindrome Checker 532
 - Quick Quiz 10.1 534
 - Exercises 10.1 534





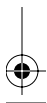
xviii Contents

10.2	Examples of Recursion: Towers of Hanoi; Parsing	538
	Towers of Hanoi	538
	Parsing	541
	Exercises 10.2	547
10.3	Implementing Recursion	548
10.4	Algorithm Efficiency	551
	Quick Quiz 10.4	562
	Exercises 10.4	563
10.5	Standard Algorithms in C++	564
	Example: STL's sort Algorithm	564
	A Sample of STL Algorithms	570
	Algorithms from the <code><numeric></code> Library	572
	Example: Judging Figure Skating	572
	Quick Quiz 10.5	573
10.6	Proving Algorithms Correct (optional)	573
	Example: Calculating the Mean	574
	Example: Recursive Power Function	576
	Summary	577
	Exercises 10.6	578
	Summary	579
	Chapter Notes	579
	Programming Pointers	579
	ADT Tips	579
	Programming Problems	580



11 More Linking Up with Linked Lists 585

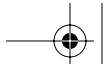
11.1	Some Variants of Singly-Linked Lists	586
	Linked Lists with Head Nodes	586
	Circular Linked Lists	589
	Quick Quiz 11.1	591
	Exercises 11.1	592
11.2	Linked Implementation of Sparse Polynomials	593
	Exercises 11.2	602
11.3	Doubly-Linked Lists and the Standard C++ <code>list</code>	602
	Doubly-Linked Lists	602
	The Standard <code>list</code> Class Template	604
	Example: Internet Addresses	612
	A Look under the Hood at C++'s <code>list</code>	616
	Exercises 11.3	620
11.4	Case Study: Large-Integer Arithmetic	621
	Problem	621
	Design	621





- Implementation 623
- Exercises 11.4 631
- 11.5 Other Multiply-Linked Lists 631
 - Multiply-Ordered Lists 631
 - Sparse Matrices 632
 - Generalized Lists 635
 - Exercises 11.5 637
- Summary 639
 - Chapter Notes 639
 - Programming Pointers 639
 - ADT Tips 639
- Programming Problems 640
- 12 Searching: Binary Trees and Hash Tables 645**
 - 12.1 Review of Linear Search and Binary Search 646
 - Linear Search 646
 - Binary Search 648
 - Exercises 12.1 650
 - 12.2 Introduction to Binary Trees 651
 - Tree Terminology 652
 - Some Examples of Binary Trees 653
 - Array Representation of Binary Trees 655
 - Linked Representation of Binary Trees 657
 - Quick Quiz 12.2 658
 - Exercises 12.2 659
 - 12.3 Binary Trees as Recursive Data Structures 660
 - Traversals 661
 - Quick Quiz 12.3 665
 - Exercises 12.3 665
 - 12.4 Binary Search Trees 667
 - Implementing BSTs 667
 - BST Traversals 670
 - Searching a BST 674
 - Inserting into a BST 677
 - Removing a Node from a BST 681
 - Problem of Lopsidedness 693
 - Quick Quiz 12.4 694
 - Exercises 12.4 695
 - 12.5 Case Study: Validating Computer Logins 698
 - Problem 698
 - Design 698
 - Coding 699





xx Contents

- 12.6 Threaded Binary Search Trees (Optional) 702
 - Quick Quiz 12.6 706
 - Exercises 12.6 706
- 12.7 Hash Tables 707
 - Hash Functions 708
 - Collision Strategies 709
 - Improvements 710
 - Quick Quiz 12.7 714
 - Exercises 12.7 714
- Summary 715
 - Chapter Notes 715
 - Programming Pointers 715
 - ADT Tips 716
- Programming Problems 716

13 Sorting 721

- 13.1 Some $O(n^2)$ Sorting Schemes 722
 - Selection Sorts 722
 - Exchange Sorts 725
 - Insertion Sorts 728
 - Evaluation of These Sorting Schemes 730
 - Indirect Sorting 731
 - Quick Quiz 13.1 732
 - Exercises 13.1 733
- 13.2 Heaps, Heapsort, and Priority Queues 735
 - Heaps 736
 - Basic Heap Operations 737
 - Heapsort 741
 - Heap Algorithms in STL 745
 - Heaps and Priority Queues 748
 - Quick Quiz 13.2 751
 - Exercises 13.2 752
- 13.3 Quicksort 753
 - The Split Operation 754
 - Quicksort 757
 - Improvements 760
 - Quick Quiz 13.3 760
 - Exercises 13.3 761
- 13.4 Mergesort 762
 - Merging Lists 762
 - Binary Mergesort 764
 - Natural Mergesort 766





- Quick Quiz 13.4 769
- Exercises 13.4 769
- 13.5 Radix Sort 770
 - Exercises 13.5 773
- Summary 774
 - Chapter Notes 774
 - Programming Pointers 774
 - ADT Tips 775
- Programming Problems 775
- 14 OOP and ADTs 779**
 - 14.1 A Brief History and Overview of OOP and ADTs 780
 - Encapsulation 781
 - Inheritance 781
 - Polymorphism and Dynamic Binding 783
 - Quick Quiz 14.1 784
 - 14.2 Inheritance and Object-Oriented Design 784
 - Example 1: Licenses 786
 - Public, Private, and Protected Sections 789
 - The Form of Derived Classes 790
 - The Is-a, Has-a, and Uses-a Relationships Between Classes 791
 - Quick Quiz 14.2 793
 - Exercises 14.2 793
 - 14.3 Building Derived Classes 794
 - Derived Class Constructors 794
 - Accessing Inherited Data Members 794
 - Reusing Operations 795
 - Example: Stacks and Bounded Stacks 797
 - 14.4 Case Study: Payroll 801
 - Problem 801
 - Design 802
 - 14.5 Polymorphism, Virtual Functions, and ADTs 810
 - Why Polymorphism is Needed: a Binding Problem 810
 - Virtual Functions and Dynamic Binding 812
 - Example 1: Using Handles 816
 - Example 2: Stacks and Bounded Stacks 817
 - Pure Virtual Functions and Abstract Classes 820
 - Quick Quiz 14.5 822
 - Exercises 14.5 823
 - 14.6 Case Study: A Heterogeneous Data Structure 823
 - The Need for Virtual Functions 824





xxii Contents

Summary	828
Chapter Notes	828
Programming Pointers	828
ADT Tips	828
Programming Problems	829

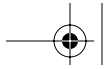
15 Trees 831

15.1 Case Study: Huffman Codes	832
Variable-Length Codes	832
Immediate Decodability	833
Huffman Codes	833
Exercises 15.1	838
15.2 Tree Balancing: AVL Trees	839
Example: A BST of State Abbreviations	840
The Basic Rebalancing Rotations	849
Quick Quiz 15.2	854
Exercises 15.2	854
15.3 2-3-4 Trees, Red-Black Trees, B-Trees, and Other Trees	855
2-3-4 Trees	856
Red-Black Trees	863
B-Trees	868
Representing Trees and Forests as Binary Trees	869
Quick Quiz 15.3	872
Exercises 15.3	873
15.4 Associative Containers in STL—maps (optional)	874
Quick Quiz 15.4	880
Summary	880
Chapter Notes	880
Programming Pointers	880
ADT Tips	880
Programming Problems	881

16 Graphs and Digraphs 883

16.1 Directed Graphs	883
Adjacency-Matrix Representation	885
Adjacency-List Representation	887
Quick Quiz 16.1	888
Exercises 16.1	889
16.2 Searching and Traversing Digraphs	892
Depth-First Search	894
Breadth-First Search	896
Traversal and Shortest-Path Problems	897





NP-Complete Problems	908
Quick Quiz 16.2	909
Exercises 16.2	909
16.3 Graphs	911
Adjacency-Matrix and Adjacency-List Representations	912
Edge-List Representation	913
Connectedness	913
Quick Quiz 16.3	922
Exercises 16.3	922
Summary	926
Chapter Notes	926
Programming Pointers	927
ADT Tips	927
Programming Problems	927

Appendixes

A ASCII Character Set A1

B Number Systems B1

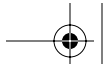
C Basic C++ C1

C.1 C++ Program Structure	C1
C.2 Compiler Directives	C2
C.3 Standard Libraries	C2
C.4 Comments	C5
C.5 Identifiers and Keywords	C6
C.6 Fundamental Data Types	C9
C.7 Literals	C9
C.8 Declarations	C10
C.9 Operators and Expressions	C10
C.10 Control Structures	C16
C.11 Functions	C22
C.12 Lifetimes, Scopes, and Namespaces	C28
C.13 Files	C30

D Other C++ Features D1

D.1 Stream Operations	D1
D.2 String Operations	D5
D.3 Exceptions	D14





xxiv Contents

D.4 More About Function Templates D17

D.5 Other Applications of Pointers D19

E From Java to C++ E1

E.1 Program Structure E1

E.2 using and Compiler Directives E1

E.3 Input and Output (§C.9, §C.13, & §D.1) E2

E.4 Data Types E5

E.5 Variables and Constants E6

E.6 Functions E7

E.7 Operator Overloading E9

E.8 Things the Same in Both C++ and Java E10

E.9 More Differences Between C++ and Java E11

F Answers to Quick Quizzes F1

Index I1

