

Data Structures in C++ Using the Standard Template Library

Timothy Budd
Oregon State University



An imprint of Addison Wesley Longman, Inc.

Reading, Massachusetts • Harlow, England • Menlo Park, California
Berkeley, California • Don Mills, Ontario • Sydney
Bonn • Amsterdam • Tokyo • Mexico City

CONTENTS

Preface v

I FUNDAMENTAL TOOLS 1

1 Fundamentals 3

- 1.1 The Study of Data Structures 3
 - 1.1.1 The STL 4
- 1.2 Language Fundamentals 4
 - 1.2.1 Comments 5
 - 1.2.2 Constants 5
 - 1.2.3 Basic Data Types and Declaration Statements 6
 - 1.2.4 Expressions and Assignment Statements 7
 - 1.2.5 Input and Output 9
 - 1.2.6 Pointer Values 9
 - 1.2.7 Conditional Statements 10
 - 1.2.8 Loops 11
 - 1.2.9 Arrays 12
 - 1.2.10 Structures 14
 - 1.2.11 Functions 14
 - 1.2.12 The Function main 16
 - 1.2.13 Include Files 17
 - 1.2.14 Binding Times 17
- 1.3 Chapter Summary 19
 - Further Reading 20
 - Study Questions & Exercises 20

2 Classes and Object-Oriented Programming 23

- Chapter Overview 23
- 2.1 The Card Game WAR 24
- 2.2 The Class Card 24
- 2.3 The Class Deck 29
 - 2.3.1 In-Line Function Definitions 34

- 2.4 The Class Player 34
- 2.5 The Game Itself 37
- 2.6 Making an Interactive Game 38
- 2.7 Accessor and Mutator Functions 39
- 2.8 Chapter Summary 40
 - Further Reading 41
 - Study Questions & Exercises 42
- 3 Algorithms: Descriptions of Behavior 45**
 - Chapter Overview 45
 - 3.1 Properties of Algorithms 46
 - 3.2 Recipes as Algorithms 47
 - 3.3 Analyzing Computer Algorithms 48
 - 3.3.1 Specification of the Input 49
 - 3.3.2 Description of the Result 51
 - 3.3.3 Instruction Precision 52
 - 3.3.4 Time to Execute 53
 - 3.3.5 Space Utilization 56
 - 3.4 Recursive Algorithms 56
 - 3.5 Chapter Summary 60
 - Further Reading 60
 - Study Questions & Exercises 61
- 4 Analyzing Execution Time 63**
 - Chapter Overview 63
 - 4.1 Algorithmic Analysis and Big-Oh Notation 64
 - 4.2 Execution Time of Programming Constructs 65
 - 4.2.1 Constant Time 65
 - 4.2.2 Simple Loops 66
 - 4.2.3 Nested Loops 68
 - 4.2.4 While Loops 71
 - 4.2.5 Function Calls 73
 - 4.2.6 Recursive Calls 74
 - 4.3 Summing Algorithmic Execution Times 76
 - 4.4 Benchmarking Actual Execution Times 80
 - 4.5 Chapter Summary 83

Further Reading 83
Study Questions & Exercises 84

5 Increasing Confidence in Correctness 87

Chapter Overview 87

- 5.1 Program Proofs 88
 - 5.1.1 Invariants 88
 - 5.1.2 Analyzing Loops 90
 - 5.1.3 Asserting the Outcome is Correct 93
 - 5.1.4 Progress Toward an Objective 94
 - 5.1.5 Manipulating Unnamed Quantities 95
 - 5.1.6 Function Calls 96
 - 5.1.7 Recursive Algorithms 97
- 5.2 Program Testing 99
- 5.3 Chapter Summary 101
 - Further Reading 101
 - Study Questions & Exercises 101

II THE STANDARD CONTAINERS 105

6 The Standard Library Container Classes 107

Chapter Overview 107

- 6.1 Container Classes 108
 - 6.1.1 Vectors 108
 - 6.1.2 Strings 110
 - 6.1.3 Lists 110
 - 6.1.4 Double-Ended Queues 110
 - 6.1.5 Stacks and Queues 111
 - 6.1.6 Sets 112
 - 6.1.7 Priority Queues 112
 - 6.1.8 Maps (Dictionaries) 113
- 6.2 Selecting a Container 113
- 6.3 Iterators 115
 - Further Reading 119
- 6.4 Chapter Summary 119
 - Study Questions & Exercises 119

7 The string Data Type 121

Chapter Overview 121

7.1	The String Data Abstraction	122
7.1.1	Include Files	122
7.1.2	Primitive (C-style) Strings	122
7.2	Problem Solving with Strings	125
7.2.1	Palindrome Testing	125
7.2.2	Split a Line into Words	129
7.3	String Operations	130
7.3.1	Declaring String Variables	131
7.3.2	Character Access	131
7.3.3	Extent of String	133
7.3.4	Assignment and Append	133
7.3.5	Iterators	134
7.3.6	Insertion, Removal, and Replacement	134
7.3.7	String Comparisons	134
7.3.8	Searching Operations	134
7.3.9	Useful Generic Algorithms	135
7.3.10	Input/Output Routines	136
7.4	The Implementation of Strings	137
7.4.1	Constructors, Assignment	139
7.4.2	Destructor and Delete	141
7.4.3	Resize Internal Buffer	142
7.4.4	Computing Length	144
7.4.5	Character Access	145
7.4.6	Iterators	146
7.4.7	Insertion, Removal, Replacement, and Append	147
7.4.8	Comparison Operators	149
7.4.9	Substring Matching	150
	Further Reading	151
7.5	Chapter Summary	151
	Study Questions & Exercises	151
8	Vectors: A Random Access Data Structure	155
	Chapter Overview	155
8.1	The Vector Data Abstraction	156
8.2	Templates	156
8.2.1	Function Templates	159
8.3	Problem Solving with Vectors	159
8.3.1	Sieve of Eratosthenes	160
8.3.2	Sorting	161

8.3.3	Merge Sort	163
8.3.4	Silly Sentence Generation*	166
8.3.5	Matrices*	168
8.4	Summary of Vector Operations	169
8.4.1	Declaration and Initialization of Vectors	169
8.4.2	Subscripting a Vector	171
8.4.3	Element Insertion	171
8.4.4	Element Removal	172
8.4.5	Extent and Size-Changing Operations	172
8.4.6	Iterators	173
8.4.7	Generic Algorithms	173
8.4.8	Sorting and Sorted Vector Operations	175
8.5	The Implementation of Vector	176
8.5.1	Constructors	176
8.5.2	Reserve and Resize	179
8.5.3	Access to Data Values	179
8.6	Implementing Generic Algorithms	180
8.7	Chapter Summary	182
	Study Questions & Exercises	182
9	Lists: A Dynamic Data Structure	185
	Chapter Overview	185
9.1	The List Data Abstraction	186
9.2	Summary of List Operations	187
9.2.1	Insert Iterators	191
9.3	Example Programs	193
9.3.1	An Inventory System	193
9.3.2	A Course Registration System	196
9.4	An Example Implementation	203
9.4.1	List Iterators	206
9.5	Variation Through Inheritance	209
9.5.1	Ordered Lists	209
9.5.2	Self-Organizing Lists	211
9.5.3	Private and Protected	212
9.6	Chapter Summary	214
	Further Reading	214
	Study Questions & Exercises	215

- 10 Stacks and Queues 217**
 - Chapter Overview 217
 - 10.1 The Stack and Queue Data Abstractions 217
 - 10.2 Adaptors 220
 - 10.3 Stacks 220
 - 10.3.1 Application: RPN Calculator 222
 - 10.3.2 Application: Conversion of Infix to Postfix* 225
 - 10.4 Queues 228
 - 10.4.1 Example Program: Bank Teller Simulation 228
 - 10.4.2 Ring Buffer Queues 231
 - 10.5 Chapter Summary 234
 - Further Reading 235
 - Study Questions & Exercises 235

- 11 Deques: Double-Ended Data Structures 237**
 - Chapter Overview 237
 - 11.1 The Deque Abstraction 238
 - 11.2 Application: Depth- and Breadth-First Search 239
 - 11.3 Application: A Framework for Backtracking 248
 - 11.3.1 Specialization Using Inheritance 250
 - 11.4 An Implementation 256
 - 11.4.1 Deque Iterators 259
 - 11.5 Chapter Summary 261
 - Study Questions & Exercises 262

- 12 Sets and Multisets 263**
 - Chapter Overview 263
 - 12.1 The Set Data Abstraction 263
 - 12.2 Set Operations 264
 - 12.3 Bit Vector Sets* 266
 - 12.4 The set Data Type 272
 - 12.4.1 A Spelling Checker 272
 - 12.4.2 Spelling Correction 274
 - 12.4.3 Anagrams 275
 - 12.5 Summary of Operations for Class set 276

12.5.1	Generic Functions for Set Operations	278
12.6	An Implementation of Class set	280
12.6.1	Implementation of Generic Algorithms	288
12.7	Chapter Summary	290
	Further Reading	290
	Study Questions & Exercises	291
13	Trees: A Nonlinear Data Structure	293
	Chapter Overview	293
13.1	Properties of Trees	294
13.2	Binary Trees	298
13.2.1	Vector Implementation	303
13.2.2	Dynamic Memory Implementation	303
13.2.3	Application: "Guess the Animal" Game	304
13.3	Operator Precedence Parsing*	307
13.4	Tree Traversals	311
13.4.1	Postorder Tree Traversal Iterator	315
13.4.2	Preorder Tree Traversal Iterator	317
13.4.3	Inorder Tree Traversal Iterator	318
13.4.4	Level-Order Tree Traversal Iterator	320
13.5	Binary Tree Representation of General Trees	321
13.6	Chapter Summary	323
	Study Questions & Exercises	323
14	Searching	327
	Chapter Overview	327
14.1	Divide and Conquer	328
14.1.1	Binary Search	329
14.1.2	Application: Root Finding	331
14.2	Ordered Vectors	332
14.3	Balanced Binary Search Trees	336
14.4	Application: Tree Sort	346
14.5	Finding the <i>N</i> th Largest	347
14.5.1	Application: Quick Sort	352
14.6	Chapter Summary	355
	Further Reading	356

Study Questions & Exercises 356

15 Priority Queues 359

Chapter Overview 359

15.1 The Priority Queue Data Abstraction 360

15.2 Heaps 362

15.2.1 Application: Heap Sort 366

15.3 Skew Heaps* 370

15.4 Application: Discrete Event-Driven Simulation 376

15.4.1 A Framework for Simulations 378

15.5 Chapter Summary 383

Further Reading 383

Study Questions & Exercises 384

16 Maps and Multimaps 387

Chapter Overview 387

16.1 The map Data Abstraction 388

16.1.1 Pairs 388

16.2 Example Programs 388

16.2.1 A Telephone Directory 388

16.2.2 Silly Sentence Generation Revisited 392

16.2.3 A Concordance 396

16.3 Operations on Maps 398

16.3.1 Include Files 398

16.3.2 Creation and Initialization 399

16.3.3 Type Definitions 400

16.3.4 Insertion and Access 400

16.3.5 Removal of Values 401

16.3.6 Iterators 401

16.3.7 Searching and Counting 401

16.4 An Example Implementation 402

16.5 Chapter Summary 404

Further Reading 405

Study Questions & Exercises 405

III OTHER CONTAINERS 407

17 Hash Tables 409

Chapter Overview 409

17.1 The Hash Table Abstraction 409

17.2 Hash Functions 410

17.3 Collision Resolution Using Buckets 413

17.3.1 Asymptotic Analysis of Hash Table Operations 413

17.4 Hash Table Sorting Algorithms 414

17.4.1 Counting Sort 414

17.4.2 Bucket Sorting 417

17.4.3 Radix Sorting 418

17.5 The hash_table Data Type 421

17.6 Hash Functions 423

17.7 Chapter Summary 425

Further Reading 426

Study Questions & Exercises 426

18 Matrices: Two-Dimensional Data Structures 429

Chapter Overview 429

18.1 The Matrix Data Abstraction 429

18.1.1 C++ Matrices 430

18.2 Matrices as Vectors of Vectors 433

18.2.1 Combining Matrices and Vectors 435

18.3 Sparse Matrices 436

18.4 Non-Integer Index Values 439

18.5 Chapter Summary 442

Study Questions & Exercises 442

19 Graphs 445

Chapter Overview 445

19.1 The Graph Data Abstraction 445

19.2 Adjacency Matrix Representation 447

19.2.1 Warshall's Algorithm 448

19.3 Edge List Representation 451

19.3.1	Reachability Using Depth-First Search	453
19.4	Weighted Adjacency Matrix	455
19.4.1	Floyd's Algorithm	457
19.5	Sparse Matrix Representation	457
19.5.1	Dijkstra's Algorithm	459
19.6	Finite Automata	463
19.7	Turing Machines*	467
19.8	Chapter Summary	471
	Further Reading	471
	Study Questions & Exercises	472

20 Files: External Collections 475

	Chapter Overview	475
20.1	The File Data Abstraction	475
20.2	Character Stream Operations	476
20.2.1	Streams and Iterators	477
20.3	Application: Lexical Analysis	479
20.4	Application: File Merge Sort	481
20.5	Binary Files	485
20.5.1	Open Table Hashing	488
20.5.2	Application: A Simple Database	493
20.6	Chapter Summary	495
	Further Reading	495
	Study Questions & Exercises	495

IV APPENDICES 497

Appendix A Common Implementation Difficulties 499

A.1	The Class <code>bool</code>	499
A.2	Integrating the <code>string</code> Library and the Container Classes	500
A.3	Optional Template Arguments	500
A.4	Allocators	501
A.5	Template Arguments on Member Functions	501

Appendix B Summary of Standard Container Operations 503

- B.1 The Standard Containers 503
 - B.1.1 String 503
 - B.1.2 Vector 504
 - B.1.3 List 505
 - B.1.4 Stack 506
 - B.1.5 Queue 506
 - B.1.6 Deque 507
 - B.1.7 Bit set 508
 - B.1.8 Set 508
 - B.1.9 Priority Queue 509
 - B.1.10 Map 509
- B.2 Generic Algorithms 510
 - B.2.1 Initialization Algorithms 510
 - B.2.2 Searching Operations 511
 - B.2.3 In-Place Transformations 512
 - B.2.4 Removal Algorithms 513
 - B.2.5 Scalar-Producing Algorithms 514
 - B.2.6 Sequence-Generating Algorithms 515
 - B.2.7 Sorting Algorithms 515
 - B.2.8 Binary Search 516
 - B.2.9 Set Operations 516
 - B.2.10 Heap Operations 517
 - B.2.11 Miscellaneous Algorithms 517

Appendix C Tables of Various Functions 519**Appendix D If C++ Is the Solution, Then What Is the Problem? 523**

- D.1 Class Definitions 524
- D.2 Member Functions 524
- D.3 Access Controls (Public, Private, and Protected) 525
- D.4 References and Pointers 525
- D.5 Reference Parameters 526
- D.6 Constructors 526
- D.7 Copy Constructor 527
- D.8 Operators as Functions and Operators as Members 528
- D.9 Conversion Operators 528
- D.10 Automatic and Dynamic Allocation 529
- D.11 Destructors 530

D.12 Friends 530

D.13 Derived Classes: Inheritance 531

D.14 Templates 531

D.15 Polymorphism: Virtual Functions 531

Bibliography 533

Index 539