

Data Structures and Program Design in C++

Robert L. Kruse
Alexander J. Ryba

CD-ROM prepared by
Paul A. Mailhot



Prentice Hall
Upper Saddle River, New Jersey 07458

Contents

Preface	ix	
Synopsis	xii	
Course Structure	xiv	
Supplementary Materials	xv	
Book Production	xvi	
Acknowledgments	xvi	
1 Programming Principles		1
1.1 Introduction	2	
1.2 The Game of Life	4	
1.2.1 Rules for the Game of Life	4	
1.2.2 Examples	5	
1.2.3 The Solution: Classes, Objects, and Methods	7	
1.2.4 Life: The Main Program	8	
1.3 Programming Style	10	
1.3.1 Names	10	
1.3.2 Documentation and Format	13	
1.3.3 Refinement and Modularity	15	
1.4 Coding, Testing, and Further Refinement	20	
1.4.1 Stubs	20	
1.4.2 Definition of the Class Life	22	
1.4.3 Counting Neighbors	23	
1.4.4 Updating the Grid	24	
1.4.5 Input and Output	25	
1.4.6 Drivers	27	
1.4.7 Program Tracing	28	
1.4.8 Principles of Program Testing	29	
1.5 Program Maintenance	34	
1.5.1 Program Evaluation	34	
1.5.2 Review of the Life Program	35	
1.5.3 Program Revision and Redevelopment	38	
1.6 Conclusions and Preview	39	
1.6.1 Software Engineering	39	
1.6.2 Problem Analysis	40	
1.6.3 Requirements Specification	41	
1.6.4 Coding	41	
Pointers and Pitfalls	45	
Review Questions	46	
References for Further Study	47	
C++	47	
Programming Principles	47	
The Game of Life	47	
Software Engineering	48	
2 Introduction to Stacks		49
2.1 Stack Specifications	50	
2.1.1 Lists and Arrays	50	
2.1.2 Stacks	50	
2.1.3 First Example: Reversing a List	51	
2.1.4 Information Hiding	54	
2.1.5 The Standard Template Library	55	

2.2 Implementation of Stacks	57
2.2.1 Specification of Methods for Stacks	57
2.2.2 The Class Specification	60
2.2.3 Pushing, Popping, and Other Methods	61
2.2.4 Encapsulation	63
2.3 Application: A Desk Calculator	66
2.4 Application: Bracket Matching	69
2.5 Abstract Data Types and Their Implementations	71
2.5.1 Introduction	71
2.5.2 General Definitions	73
2.5.3 Refinement of Data Specification	74
Pointers and Pitfalls	76
Review Questions	76
References for Further Study	77

3 Queues _____ 78

3.1 Definitions	79
3.1.1 Queue Operations	79
3.1.2 Extended Queue Operations	81
3.2 Implementations of Queues	84
3.3 Circular Implementation of Queues in C++	89
3.4 Demonstration and Testing	93
3.5 Application of Queues: Simulation	96
3.5.1 Introduction	96
3.5.2 Simulation of an Airport	96
3.5.3 Random Numbers	99
3.5.4 The Runway Class Specification	99
3.5.5 The Plane Class Specification	100
3.5.6 Functions and Methods of the Simulation	101
3.5.7 Sample Results	107
Pointers and Pitfalls	110
Review Questions	110
References for Further Study	111

4 Linked Stacks and Queues _____ 112

4.1 Pointers and Linked Structures	113
4.1.1 Introduction and Survey	113
4.1.2 Pointers and Dynamic Memory in C++	116
4.1.3 The Basics of Linked Structures	122
4.2 Linked Stacks	127
4.3 Linked Stacks with Safeguards	131
4.3.1 The Destructor	131
4.3.2 Overloading the Assignment Operator	132
4.3.3 The Copy Constructor	135
4.3.4 The Modified Linked-Stack Specification	136
4.4 Linked Queues	137
4.4.1 Basic Declarations	137
4.4.2 Extended Linked Queues	139
4.5 Application: Polynomial Arithmetic	141
4.5.1 Purpose of the Project	141
4.5.2 The Main Program	141
4.5.3 The Polynomial Data Structure	144
4.5.4 Reading and Writing Polynomials	147
4.5.5 Addition of Polynomials	148
4.5.6 Completing the Project	150
4.6 Abstract Data Types and Their Implementations	152
Pointers and Pitfalls	154
Review Questions	155

5 Recursion _____ 157

5.1 Introduction to Recursion	158
5.1.1 Stack Frames for Subprograms	158
5.1.2 Tree of Subprogram Calls	159
5.1.3 Factorials: A Recursive Definition	160
5.1.4 Divide and Conquer: The Towers of Hanoi	163
5.2 Principles of Recursion	170
5.2.1 Designing Recursive Algorithms	170
5.2.2 How Recursion Works	171
5.2.3 Tail Recursion	174
5.2.4 When Not to Use Recursion	176
5.2.5 Guidelines and Conclusions	180

5.3 Backtracking: Postponing the Work	183
5.3.1 Solving the Eight-Queens Puzzle	183
5.3.2 Example: Four Queens	184
5.3.3 Backtracking	185
5.3.4 Overall Outline	186
5.3.5 Refinement: The First Data Structure and Its Methods	188
5.3.6 Review and Refinement	191
5.3.7 Analysis of Backtracking	194
5.4 Tree-Structured Programs: Look-Ahead in Games	198
5.4.1 Game Trees	198
5.4.2 The Minimax Method	199
5.4.3 Algorithm Development	201
5.4.4 Refinement	203
5.4.5 Tic-Tac-Toe	204
Pointers and Pitfalls	209
Review Questions	210
References for Further Study	211

6 Lists and Strings _____ 212

6.1 List Definition	213
6.1.1 Method Specifications	214
6.2 Implementation of Lists	217
6.2.1 Class Templates	218
6.2.2 Contiguous Implementation	219
6.2.3 Simply Linked Implementation	221
6.2.4 Variation: Keeping the Current Position	225
6.2.5 Doubly Linked Lists	227
6.2.6 Comparison of Implementations	230
6.3 Strings	233
6.3.1 Strings in C++	233
6.3.2 Implementation of Strings	234
6.3.3 Further String Operations	238
6.4 Application: A Text Editor	242
6.4.1 Specifications	242
6.4.2 Implementation	243
6.5 Linked Lists in Arrays	251
6.6 Application: Generating Permutations	260
Pointers and Pitfalls	265
Review Questions	266
References for Further Study	267

7 Searching _____ 268

7.1 Searching: Introduction and Notation	269
7.2 Sequential Search	271
7.3 Binary Search	278
7.3.1 Ordered Lists	278
7.3.2 Algorithm Development	280
7.3.3 The Forgetful Version	281
7.3.4 Recognizing Equality	284
7.4 Comparison Trees	286
7.4.1 Analysis for $n = 10$	287
7.4.2 Generalization	290
7.4.3 Comparison of Methods	294
7.4.4 A General Relationship	296
7.5 Lower Bounds	297
7.6 Asymptotics	302
7.6.1 Introduction	302
7.6.2 Orders of Magnitude	304
7.6.3 The Big-O and Related Notations	310
7.6.4 Keeping the Dominant Term	311
Pointers and Pitfalls	314
Review Questions	315
References for Further Study	316

8 Sorting _____ 317

8.1 Introduction and Notation	318
8.1.1 Sortable Lists	319
8.2 Insertion Sort	320
8.2.1 Ordered Insertion	320
8.2.2 Sorting by Insertion	321
8.2.3 Linked Version	323
8.2.4 Analysis	325
8.3 Selection Sort	329
8.3.1 The Algorithm	329
8.3.2 Contiguous Implementation	330
8.3.3 Analysis	331
8.3.4 Comparisons	332
8.4 Shell Sort	333
8.5 Lower Bounds	336

8.6 Divide-and-Conquer Sorting	339
8.6.1 The Main Ideas	339
8.6.2 An Example	340
8.7 Mergesort for Linked Lists	344
8.7.1 The Functions	345
8.7.2 Analysis of Mergesort	348
8.8 Quicksort for Contiguous Lists	352
8.8.1 The Main Function	352
8.8.2 Partitioning the List	353
8.8.3 Analysis of Quicksort	356
8.8.4 Average-Case Analysis of Quicksort	358
8.8.5 Comparison with Mergesort	360
8.9 Heaps and Heapsort	363
8.9.1 Two-Way Trees as Lists	363
8.9.2 Development of Heapsort	365
8.9.3 Analysis of Heapsort	368
8.9.4 Priority Queues	369
8.10 Review: Comparison of Methods	372
Pointers and Pitfalls	375
Review Questions	376
References for Further Study	377
9 Tables and Information Retrieval	379
9.1 Introduction:	
Breaking the $\lg n$ Barrier	380
9.2 Rectangular Tables	381
9.3 Tables of Various Shapes	383
9.3.1 Triangular Tables	383
9.3.2 Jagged Tables	385
9.3.3 Inverted Tables	386
9.4 Tables: A New Abstract Data Type	388
9.5 Application: Radix Sort	391
9.5.1 The Idea	392
9.5.2 Implementation	393
9.5.3 Analysis	396
9.6 Hashing	397
9.6.1 Sparse Tables	397
9.6.2 Choosing a Hash Function	399
9.6.3 Collision Resolution with Open Addressing	401
9.6.4 Collision Resolution by Chaining	406
9.7 Analysis of Hashing	411
9.8 Conclusions:	
Comparison of Methods	417
9.9 Application:	
The Life Game Revisited	418
9.9.1 Choice of Algorithm	418
9.9.2 Specification of Data Structures	419
9.9.3 The Life Class	421
9.9.4 The Life Functions	421
Pointers and Pitfalls	426
Review Questions	427
References for Further Study	428
10 Binary Trees	429
10.1 Binary Trees	430
10.1.1 Definitions	430
10.1.2 Traversal of Binary Trees	432
10.1.3 Linked Implementation of Binary Trees	437
10.2 Binary Search Trees	444
10.2.1 Ordered Lists and Implementations	446
10.2.2 Tree Search	447
10.2.3 Insertion into a Binary Search Tree	451
10.2.4 Treesort	453
10.2.5 Removal from a Binary Search Tree	455
10.3 Building a Binary Search Tree	463
10.3.1 Getting Started	464
10.3.2 Declarations and the Main Function	465
10.3.3 Inserting a Node	466
10.3.4 Finishing the Task	467
10.3.5 Evaluation	469
10.3.6 Random Search Trees and Optimality	470
10.4 Height Balance: AVL Trees	473
10.4.1 Definition	473
10.4.2 Insertion of a Node	477
10.4.3 Removal of a Node	484
10.4.4 The Height of an AVL Tree	485
10.5 Splay Trees:	
A Self-Adjusting Data Structure	490
10.5.1 Introduction	490
10.5.2 Splaying Steps	491
10.5.3 Algorithm Development	495

- 10.5.4 Amortized Algorithm Analysis: Introduction 505
- 10.5.5 Amortized Analysis of Splaying 509

Pointers and Pitfalls 515

Review Questions 516

References for Further Study 518

11 Multiway Trees 520

11.1 Orchards, Trees, and Binary Trees 521

- 11.1.1 On the Classification of Species 521
- 11.1.2 Ordered Trees 522
- 11.1.3 Forests and Orchards 524
- 11.1.4 The Formal Correspondence 526
- 11.1.5 Rotations 527
- 11.1.6 Summary 527

11.2 Lexicographic Search Trees: Tries 530

- 11.2.1 Tries 530
- 11.2.2 Searching for a Key 530
- 11.2.3 C++ Algorithm 531
- 11.2.4 Searching a Trie 532
- 11.2.5 Insertion into a Trie 533
- 11.2.6 Deletion from a Trie 533
- 11.2.7 Assessment of Tries 534

11.3 External Searching: B-Trees 535

- 11.3.1 Access Time 535
- 11.3.2 Multiway Search Trees 535
- 11.3.3 Balanced Multiway Trees 536
- 11.3.4 Insertion into a B-Tree 537
- 11.3.5 C++ Algorithms: Searching and Insertion 539
- 11.3.6 Deletion from a B-Tree 547

11.4 Red-Black Trees 556

- 11.4.1 Introduction 556
- 11.4.2 Definition and Analysis 557
- 11.4.3 Red-Black Tree Specification 559
- 11.4.4 Insertion 560
- 11.4.5 Insertion Method Implementation 561
- 11.4.6 Removal of a Node 565

Pointers and Pitfalls 566

Review Questions 567

References for Further Study 568

12 Graphs 569

12.1 Mathematical Background 570

- 12.1.1 Definitions and Examples 570
- 12.1.2 Undirected Graphs 571
- 12.1.3 Directed Graphs 571

12.2 Computer Representation 572

- 12.2.1 The Set Representation 572
- 12.2.2 Adjacency Lists 574
- 12.2.3 Information Fields 575

12.3 Graph Traversal 575

- 12.3.1 Methods 575
- 12.3.2 Depth-First Algorithm 577
- 12.3.3 Breadth-First Algorithm 578

12.4 Topological Sorting 579

- 12.4.1 The Problem 579
- 12.4.2 Depth-First Algorithm 580
- 12.4.3 Breadth-First Algorithm 581

12.5 A Greedy Algorithm:

- Shortest Paths 583**
- 12.5.1 The Problem 583
- 12.5.2 Method 584
- 12.5.3 Example 585
- 12.5.4 Implementation 586

12.6 Minimal Spanning Trees 587

- 12.6.1 The Problem 587
- 12.6.2 Method 589
- 12.6.3 Implementation 590
- 12.6.4 Verification of Prim's Algorithm 593

12.7 Graphs as Data Structures 594

Pointers and Pitfalls 596

Review Questions 597

References for Further Study 597

13 Case Study: The Polish Notation 598

13.1 The Problem 599

- 13.1.1 The Quadratic Formula 599

13.2 The Idea 601

- 13.2.1 Expression Trees 601
- 13.2.2 Polish Notation 603

13.3 Evaluation of Polish Expressions **604**

- 13.3.1 Evaluation of an Expression in Prefix Form 605
- 13.3.2 C++ Conventions 606
- 13.3.3 C++ Function for Prefix Evaluation 607
- 13.3.4 Evaluation of Postfix Expressions 608
- 13.3.5 Proof of the Program: Counting Stack Entries 609
- 13.3.6 Recursive Evaluation of Postfix Expressions 612

13.4 Translation from Infix Form to Polish Form **617**

13.5 An Interactive Expression Evaluator **623**

- 13.5.1 Overall Structure 623
- 13.5.2 Representation of the Data: Class Specifications 625
- 13.5.3 Tokens 629
- 13.5.4 The Lexicon 631
- 13.5.5 Expressions: Token Lists 634
- 13.5.6 Auxiliary Evaluation Functions 639
- 13.5.7 Graphing the Expression: The Class Plot 640
- 13.5.8 A Graphics-Enhanced Plot Class 643

References for Further Study 645

A **Mathematical Methods** **647**

A.1 Sums of Powers of Integers **647**

A.2 Logarithms **650**

- A.2.1 Definition of Logarithms 651
- A.2.2 Simple Properties 651
- A.2.3 Choice of Base 652
- A.2.4 Natural Logarithms 652
- A.2.5 Notation 653
- A.2.6 Change of Base 654
- A.2.7 Logarithmic Graphs 654
- A.2.8 Harmonic Numbers 656

A.3 Permutations, Combinations, Factorials **657**

- A.3.1 Permutations 657
- A.3.2 Combinations 657
- A.3.3 Factorials 658

A.4 Fibonacci Numbers **659**

A.5 Catalan Numbers **661**

- A.5.1 The Main Result 661
- A.5.2 The Proof by One-to-One Correspondences 662
- A.5.3 History 664
- A.5.4 Numerical Results 665

References for Further Study 665

B **Random Numbers** **667**

B.1 Introduction **667**

B.2 Strategy **668**

B.3 Program Development **669**

References for Further Study 673

C **Packages and Utility Functions** **674**

C.1 Packages and C++ Translation Units **674**

C.2 Packages in the Text **676**

C.3 The Utility Package **678**

C.4 Timing Methods **679**

D **Programming Precepts, Pointers, and Pitfalls** **681**

D.1 Choice of Data Structures and Algorithms **681**

- D.1.1 Stacks 681
- D.1.2 Lists 681
- D.1.3 Searching Methods 682
- D.1.4 Sorting Methods 682
- D.1.5 Tables 682
- D.1.6 Binary Trees 683
- D.1.7 General Trees 684
- D.1.8 Graphs 684

D.2 Recursion **685**

D.3 Design of Data Structures **686**

D.4 Algorithm Design and Analysis **687**

D.5 Programming **688**

D.6 Programming with Pointer Objects **689**

D.7 Debugging and Testing **690**

D.8 Maintenance **690**

Index **693**