Contents

Preface xxv

Chapter 1 The Fundamentals of Computer Science 1

Introduction 1

1.1 Solving Problems with the Computer 2
An Overview of Problem Solving 3
Analyzing the Problem 4
Designing a Solution 5
Implementing the Design 8
Testing the Code 8
Maintaining the Product 9
Summary 9

1.2 Breadth: The Computer Science Discipline 10
Theory Paradigm 11
Abstraction Paradigm 11
Design Paradigm 12

1.3 Model of a Computer System 12
Input and Output Devices 13
Secondary Storage 13
Central Processing Unit 14
Main Memory 15

1.4 Breadth: Invention of the First Computers 17

1.5 Steps to Execution 19
Editor 19
1.6 Breadth: The History of C and C++ 24

1.7 Implementation of the Design 25

1.8 Top-Down Design and Functions 35

1.9 Breadth: Subject Areas of Computer Science 46

Programming and Debugging Hints 53
Key Terms 55
Summary 56
Review Questions 58
Laboratory 59

Chapter 2  Integer Variables and Expressions 63

Introduction 63

2.1 Integer Data 64
Variables 64
Variable Declaration 65
Naming of Variables 66

2.2 The Assignment Statement 69
lvalues and rvalues 71
Declaration-Initialization 72
Constants 73
Not an Algebraic Formula 74
Labeled Output 75
endl 76

2.3 Integer Arithmetic 78
Four Binary Operators 78
Modulus Operator 81
Unary Minus 85
Operator Precedence 85

2.4 Breadth: Computer Storage of Integers 90
Binary Representation of Integers 90
Counting 91
Decrementing by 1 93
Range of Unsigned Integers in a Computer 94
Conversion of a Decimal Integer to Binary 95
2.5 Breadth: Integer Arithmetic 99
   Signed-Magnitude Representation 99
   Two's Complement Representation 100
   Addition 103
   Subtraction 105
   Multiplication and Division by Two 105

2.6 Interactive Programs 108
   Interactive versus Batch Programs 108
   Interactive Programs in C++ 109

Programming and Debugging Hints 112
   Clarity of Code 112
   Clarity of User Interface 112

Key Terms 113

Summary 113

Review Questions 115

Laboratory 116

Chapter 3  Integer Functions and Encapsulation 121

Introduction 121

3.1 Problem Solving with Integer Functions 122
   Preconditions and Postconditions 122
   Analysis and Design of a Function 124
   Implementation of an Integer Function 125
   Procedures 127
   Arguments and Parameters 130

3.2 Problem Solving Revisited 137
   Analysis 138

3.3 Scope and Lifetime 140
   Local Variables and Scope 140
   Pass by Value 141
Local Variables with the Same Name 143
Global Variables 145
Namespace Revisited 149

3.4 Encapsulation with Objects 151
Abstraction 151
Object-Oriented Programming 153
Information Hiding and Classes 154
Class Diagram 156
Objects 156

Programming and Debugging Hints 159
Clarity of Comments 159
Local over Global 160

Key Terms 161
Summary 162
Review Questions 164
Laboratory 164

Chapter 4 Decision Making and Classes 171

Introduction 171

4.1 Relational and Logical Operators 172
Relational Operators 173
Logical Operators 173
Boolean Constants, Expressions, and Variables 177
Operator Precedence 178

4.2 Selection 181
Flow of Control 181
The if Statement 182
The if-else Statement 185
Conditional Expression Operator 192

4.3 Nesting 195
4.4 Multiple-Way Selection 207
   The switch Statement 207
   Branching to the Same Point 216

4.5 Defining Methods 220
   Access to Data Structure 220
   Header File 221
   File of Method Definitions 222

4.6 Breadth: Logic 226
   George Boole and Edmund Berkeley 226
   Basic Components of Logic 226
   Truth Tables 227
   Algebra of Propositions 228
   DeMorgan’s Laws 230

4.7 Testing Schemes 232
   Top-Down Testing 232
   Bottom-Up Testing 237
   Combined Top-Down and Bottom-Up Testing 238
   Class Testing 239

Programming and Debugging Hints 242
   Decision Control Structures 242
   Testing 243

Key Terms 244

Summary 245

Review Questions 248

Laboratory 249

Chapter 5  More Numeric Types and Polymorphism 259

Introduction 259

5.1 Floating Point Numbers 261
   Distinctions between Integers and Floating Point Numbers 261
   Floating Point Arithmetic 262
Exponential Notation 263
Type double 264
Formatted Output 264

5.2 Polymorphism 269

5.3 Constructors 274
Definition and Use 274
Polymorphic Constructors 275

5.4 Breadth: Floating Point Number Storage 278
Conversion from Base 2 to Base 10 278
Conversion from Base 10 to Base 2 278
Multiplication and Division by 2 280
Storage of Floating Point Numbers 280
Truncation Error 283

5.5 Coercion 286
Implicit Coercion 286
Explicit Coercion 287
Strong and Weak Typing 289

5.6 Additional Integer Types 291
Different Sizes of Integers 291
Unsigned Integers 292
Mixed-Mode Arithmetic 292
Abstract Data Type Money 292
Implementation of ADT Money 293

5.7 C++ Header Files 304
Numerical Constants 304
Absolute Value Function 306
Square Root Function 307
Additional cmath Library Functions 308

Programming and Debugging Hints 320
Interfaces between Functions: Global Variables 320
Constants 321
Reader's Understanding of the Interface 321
Default Constructors 322
Key Terms 323
Summary 323
Review Questions 325
Laboratory 326

Chapter 6  Looping 333

Introduction 333

6.1 Updating Assignment Operators 334
  Increment and Decrement Operators 336
  Pre- and Post-Increment and Decrement 337

6.2 Looping with a Pretest 338
  The while Loop 338
  Infinite Loop 341
  Nature of the Pretest 343
  Manipulation of Loop Variable 344

6.3 Looping with a Posttest 352
  The do-while Loop 352
  Applications 353

6.4 Looping and Interactive Programs 363
  The Sentinel Technique 363
  Random Numbers in Interactive Programs 366
  Seeding the Random Number Generator 367
  Ranges of Random Numbers 369

6.5 Structured and Object-Oriented Programs 379
  Structured Programming 379
  Object-Oriented Programming 381

6.6 Breadth: Computer Time 382
  Clock Cycle 383
  Clock Frequency 384
  Flops 384
Chapter 8  Characters 445

Introduction 445

8.1 Character Input and Output 446
Buffers 448
Y/N Responses 454

8.2 The ASCII Encoding Scheme 459
Numeric Code 459
Integer Equivalent of Character Digit 464
Escape Sequences 464

8.3 Character Functions 468
Changing Case 468
Boolean Character Functions 472

8.4 Breadth: Octal and Hexadecimal Systems 480
Conversion to Decimal Numbers 480
Conversion between Binary and Hexadecimal Number Systems 481
Constants 482
Applications 482
Conversion of Decimal Numbers to Hexadecimal 483

Programming and Debugging Hints 486
Defensive Programming: Detection and Recovery 486
Defensive Programming: Read Data as Strings 486
Defensive Programming: “Bullet-Proof” All Levels 487

Key Terms 488
Summary 488
Review Questions 490
Laboratory 491
Chapter 9  Inheritance and Files 501

Introduction 501

9.1 Inheritance 502

9.2 File I/O Classes 505
   File I/O 506
   Reading Characters 509
   Formatted I/O 510
   Closing and Opening Files 510

9.3 Pass by Reference 512
   Pass by Value 512
   Simple Variables as Parameters 512
   Streams as Parameters 517

9.4 Breadth: Secondary Storage 520
   Tape Storage 520
   Disk Storage 523
   CD-ROM 525

9.5 Breadth: Machine and Assembler Languages 527
   Machine Architecture 527
   Machine Instructions 527
   Fetch/Execute Cycle 530
   Branching 532
   CPU Simulator Program cpusim 537

Programming and Debugging Hints 541
   Debugging with Conditional Compilation 541
   Levels of Debugging 542
   Debugging with assert 543

Key Terms 544

Summary 544

Review Questions 545
Chapter 10  Arrays and the vector Class  555

Introduction  555

10.1 What Is an Array?  556
Declaration  557
Assigning Values  558
Array Index  559
Declaration-Initialization  566

10.2 Functions with Array Parameters  568
Passing an Array  568
Minimum and Maximum  571
Frequency  574

10.3 Arrays and Classes  582
An Array of Objects  582
Arrays as Member Data  583

10.4 Sequential and Binary Searches  591
Sequential Search  591
Binary Search  594

10.5 Selection Sort  602
Selection Sort Algorithm  602
Index of Minimum Element  604
Swapping Values  604

10.6 Templates  607
Templated Functions  607
Templated Class Interface  609
10.7 Arrays and the vector Class 614
  Array Limitations 614
  Range Checking 614
  Dynamic Allocation 615
  C++ vector class 616
  Vector Size and Resize 617
  Indexing 617
  A Test Program 618

10.8 Multidimensional Arrays and Vectors 621
  Indices 621
  Declaration 623
  Multidimensional Vectors 633

10.9 Breadth: Color in Computer Graphics 640
  Display Devices 641
  Color Lookup Table 642

Programming and Debugging Hints 644
Selecting Test Data: Edge Data 644
Data to Test Branches 644
Testing by Novices or with Random Data 645

Key Terms 645
Summary 646
Review Questions 648
Laboratory 650

Chapter II Strings and the string Class 657

Introduction 657

11.1 Character Strings 658
  Literals 658
  Displaying Strings 660
Reading Strings with the Extraction Operator 661
Reading a Character or a Line 663
Two-Dimensional Array of Characters 667

11.2 Validation of Data 670

11.3 Several C String Functions 677
Storage Size 678
String Length 679
Copying Strings 679
Concatenation 680

11.4 String Comparisons 684
Comparison Function 684
Summary of String Operations 698

11.5 The string Class 701
Disadvantages of C Strings 701
Constructors 702
Overloaded Operators 702
Length 703
Substrings 703
Nonmember Functions 707
C Strings versus C++ Strings 708

11.6 Breadth: Software Life Cycle 711
Analysis 713
Design 716
Implementation and Testing 717
Maintenance 719

Programming and Debugging Hints 720
Verification 720
String Comparison 720
Semicolon and Closing Brace 720
Commenting Out 721

Key Terms 722
Summary 722
Chapter 12  Structures and Enumeration Types 737

12.1 The Concept of a Structure 738
   Declaration 738
   Reference to Members 740
   Element by Element 741
   Structure Members 741
   Arrays of Structures 742

12.2 Breadth: Databases 755
   Traditional File Processing 756
   Database Management System 757
   Relational Database 758

12.3 Enumeration Types 762
   Implementation 763
   Coercion 763

12.4 Breadth: Computer Vision 771
   The Digital Image 772
   The Low-Level Module 774
   An Example of Low-Level Processing 777
   The High-Level Module 787

Programming and Debugging Hints 791
Structures with Array Elements 791

Key Terms 792
Summary 792
Review Questions 793
Laboratory 794
Chapter 13  Recursion 801

Introduction 801

13.1 Recursive Functions 802
   Power Function 811

13.2 Recursion versus Iteration 827
   Solving a Recursive Routine 827

13.3 Breadth: Formal Grammars 831
   BNF 831
   Parsing 832

Programming and Debugging Hints 840
   Run-Time Errors and Buffering 840
   Program Correctness 840

Key Terms 841
Summary 841
Review Questions 842
Laboratory 842

Chapter 14  Pointers 853

Introduction 853

14.1 The Concept of Pointers 854
   Declarations 854
   Address Operator 855
   Indirection Operator 856
   NULL 857
   Printing Addresses 858
   Point before Dereferencing 862

14.2 Breadth: Memory 864
   Memory Sizes 865
   RAM and ROM 867
14.3 Arrays and Pointers 869
   Array Name as a Constant Pointer 869
   Parameters 871
   Pointer Arithmetic 872

14.4 Dynamic Memory Allocation 875
   Allocate Memory 875
   Deallocate Memory 876

14.5 Pointer Data Members 881
   Destructors 881
   Creating and Destroying 882
   Copy Constructor 886
   Overloading Assignment 889

14.6 User-Defined Types 894
   Type Definitions in Header Files 894

14.7 Pointers and Structures 896
   Dereferencing 896
   Linked Lists 898

14.8 Function Pointers 900
   Declarations and Assignments 901
   Calling Functions Indirectly 902
   Function Pointers as Parameters 902

Programming and Debugging Hints 908
   Pointer Post-Increment 908
   Allocation of Space 911

Key Terms 911

Summary 911

Review Questions 914

Laboratory 915

Chapter 15 Data Structures 921

Introduction 921
15.1 List Abstraction 922
Definition of ADT List 922
Picturing the Current Position 923
Picturing Insertions 926
Picturing Deletions 926
Picturing Storing and Retrieving 928
Destroy a List 929
Create a List 929
List Traversal 932

15.2 Dynamic Implementation of Lists 934
List Creation 934
Initialize List 936
Test for Empty or Full 936
Current Position at the Head or Tail 937
Advance the Current Position 938
Create a Node 939
Insert a Node 940
Delete a Node 945
Store and Retrieve Information 949
Display a List 958

15.3 Stack Abstraction 961
Definition of ADT Stack 962
Additional Operations 963

15.4 Dynamic Implementation of Stacks 971
Framework for Stack 971
Initialization and Testing of a Stack 972
Push 973
Pop 973
Destroy a Stack 974
Using the Stack Data Structure 979

15.5 Breadth: Run-Time Stack 982
Programming and Debugging Hints 986
Freeing Pointers 986
Internal Limitations 986