Contents at a Glance

Introduction 1

Part 1: Getting Started with C++ Programming 7

CHAPTER 1: Writing Your First C++ Program 9
CHAPTER 2: Storing Stuff in Variables 33
CHAPTER 3: Doing the Math 51
CHAPTER 4: Show Me the Good Stuff! 59
CHAPTER 5: Performing Logical Operations 71
CHAPTER 6: Controlling the Flow 83

Part 2: Becoming a Functional C++ Programmer 101

CHAPTER 7: Creating Functions 103
CHAPTER 8: Grouping Similar Things Together Using Arrays 123
CHAPTER 9: Taking a First Look at C++ Pointers 149
CHAPTER 10: Taking a Second Look at C++ Pointers 167
CHAPTER 11: Using the C++ Preprocessor 183

Part 3: Giving Your Program a Bit of Class 207

CHAPTER 12: Examining Object-Oriented Programming 209
CHAPTER 13: Adding Class to C++ 215
CHAPTER 14: Separating Letters from Words: Character Arrays versus Strings 235
CHAPTER 15: Pointing and Staring at Objects 251
CHAPTER 16: Protecting Members: Do Not Disturb 275
CHAPTER 17: "Why Do You Build Me Up, Just to Tear Me Down, Baby?" 287
CHAPTER 18: Making Constructive Arguments 301
CHAPTER 19: Making Copies with the Copy/Move Constructor 327
CHAPTER 20: Adding Static Members: Can Fabric Softener Help? 345

Part 4: A First Look at Inheritance 359

CHAPTER 21: Passing the DNA: Sharing Code with Inheritance 361
CHAPTER 22: Creating Virtual Member Functions: Are They for Real? 371
CHAPTER 23: Factoring Classes 383

Part 5: Object-Oriented Programming in Overdrive 395

CHAPTER 24: Adopting a New Assignment Operator, Should You Decide to Accept It 397
CHAPTER 25: Playing with the Computer's File System 413

COPYRIGHTED MATERIAL

CHAPTER 26: Twice the Fun: Tapping into Multiple Inheritance 421
CHAPTER 27: Getting Ahead of Problems: Exception Handling, Contracts, and More 435

Part 6: The Part of Tens 455
CHAPTER 28: Ten Ways to Avoid Adding Bugs to Your Program 457
CHAPTER 29: Ten Ways to Make Your Programs Easier to Update and Understand 465

Index 473

Table of Contents

INTRODUCTION 1
About This Book 1
Conventions Used in this Book 2
Be Aware: This Is a Modern C++ Book 3
Icons Used in This Book 4
Beyond the Book 5
Where to Go from Here 6

PART 1: GETTING STARTED WITH C++ PROGRAMMING 7

CHAPTER 1: Writing Your First C++ Program 9
Grasping C++ Concepts 10
Installing Code::Blocks 11
Microsoft Windows 12
Ubuntu Linux 15
Macintosh 17
Creating Your First C++ Program 17
Creating a project 18
Entering the C++ code 20
Cheating 22
Building your program 23
Executing Your Program 24
Reviewing the Annotated Program 25
Examining the framework for all C++ programs 25
Clarifying source code with comments 27
Basing programs on C++ statements 28
Writing declarations 29
Generating output 29
Calculating Expressions 30
Storing the results of an expression 31
Examining the remainder of Conversion 31
And Now, a Few Comments on C++26 32

CHAPTER 2: Storing Stuff in Variables 33
Declaring Variables 34
Declaring Different Types of Variables 34
Not all numbers are the same 35
Reviewing the limitations of integers in C++ 35
Solving the truncation problem 37
Looking at the limits of floating-point numbers 38

Declaring Variable Types 39
Types of constants. 42
Range of numeric types 43
Special characters 44
Carrying Wide Loads on the char Highway 46
Automatic Declarations 49

CHAPTER 3: **Doing the Math** 51
Performing Simple Binary Arithmetic. 52
Decomposing Expressions. 54
Determining the Order of Operations 54
Performing Unary Operations. 55
Using Assignment Operators. 57

CHAPTER 4: **Show Me the Good Stuff!** 59
The Old versus the New 59
Printing the Old Way 60
A note about streaming 62
While I'm at it: Getting user input 62
Printing special characters 63
Printing the Modern Way 64
Simple printing. 64
Printing variables. 65
Adding a little bit of formatting 66
The Pros and Cons of the Old Ways Versus the New. 69
Make It Stop! 69

CHAPTER 5: **Performing Logical Operations** 71
Why Mess with Logical Operations? 71
Using the Simple Comparison Operators 72
Storing logical values. 73
Using logical int variables. 74
Performing logical operations (carefully!) on floating-point variables 75
Adding Logic with the Logical Operators 76
Making complex decisions 76
Short circuits and C++ 77
Streamlining with the Three-Way Comparison Operator 78
Peeking at the Bitwise Logical Operations. 79

CHAPTER 6: **Controlling the Flow**. 83
Controlling Program Flow with the Branch Commands 84
Executing Loops in a Program. 86
Looping while a condition is true 86

Using the autoincrement/autodecrement feature 88
Using the for loop 89
Avoiding the dreaded infinite loop 92
For each his own 93
Applying special loop controls 94
Nesting Control Commands 97
Switching to a Different Subject? 98

PART 2: BECOMING A FUNCTIONAL C++ PROGRAMMER 101

CHAPTER 7: Creating Functions 103
Writing and Using a Function 104
Exploring Functions 105
Choosing not to respond 106
Understanding simple functions 107
Writing functions 108
Understanding Functions with Arguments 111
Functions with arguments 111
Functions with multiple arguments 111
main() exposed 112
Overloading Function Names 112
Defining Function Prototypes 115
Defaulting Arguments 117
Passing by Value and Passing by Reference 118
Understanding a Variable's Visibility 120
Benefiting from Functions 121

CHAPTER 8: Grouping Similar Things Together Using Arrays 123
Making the Case for Arrays 124
Using an Array 125
Initializing an array 129
Accessing too far into an array 130
Arraying range-based for loops 131
A safer way to use arrays 132
Using Arrays of Characters 134
Creating an array of characters 134
Creating a string of characters 135
Manipulating Strings with Character 137
Defining and Using Arrays of Arrays 139
Using the Array Library Functions 141
And Now the Bad News, Which Is Actually Good News 143
Buffer Overflow: Overfilling Your Arrays 143

Avoiding buffer overflow 144
Checking out a better way to avoid buffer overflow 145
Making Room for Wide Strings 146

CHAPTER 9: **Taking a First Look at C++ Pointers** 149
What's in an Address? 150
Addressing Address Operators 150
Using Pointer Variables 151
Passing Pointers to Functions 152
Passing by value 153
Passing pointer values 154
Passing by reference 155
Constant const Irritation 156
Returning a Pointer from a Function 158
Defining limited scope 158
Examining the scope problem 160
Using Pointers and Allocating Memory for Variables 160
Making Life Safe with Smart Pointers 161
Pointing to an array 163
Keeping your pointer to yourself 163
Using New Even Though It's Now Old 164

CHAPTER 10: **Taking a Second Look at C++ Pointers** 167
Performing Operations on Pointer Variables 167
Reexamining arrays in light of pointer variables 168
Applying operators to the address of an array 170
Expanding pointer operations to a string 172
Applying operators to pointer types other than char 174
Contrasting a pointer with an array 174
Declaring and Using Arrays of Pointers 176
Utilizing arrays of character strings 177
Accessing the arguments to main() 179
Accessing program arguments, DOS-style 180

CHAPTER 11: **Using the C++ Preprocessor** 183
What Is a Preprocessor? 184
#Defining Things 185
Okay, how about not defining things? 188
A better way to define things 188
Enumerating other options 189
Including Things #if I Say So 192
Intrinsically Defined Objects 194

Using Using and Typedef 196
#Including Files 197
Organizing with Modules 201
Why modules are better 202
Creating a module 202
Using a module 204
Compiling your app that uses a module 205
One More Concept to Cover 205

PART 3: GIVING YOUR PROGRAM A BIT OF CLASS 207

CHAPTER 12: Examining Object-Oriented Programming 209
Abstracting Microwave Ovens 209
Preparing functional nachos 210
Preparing object-oriented nachos 211
Classifying Microwave Ovens 211
Why Classify? 212
A Few Other Thoughts on OOP 213

CHAPTER 13: Adding Class to C++ 215
Formatting a Class 216
Accessing the Members of a Class 217
Activating Our Objects 218
Simulating real-world objects 218
Why bother with member functions? 219
Adding a Member Function 220
Calling a Member Function 221
Accessing Other Members from a Member Function 223
Scope Resolution (and I Don't Mean How Well Your Telescope Works) 225
Defining a Member Function in the Class 226
Keeping a Member Function after Class 229
Overloading Member Functions 230
Holding a Class in Public: Using Structs 231
Nesting Structs and Classes 232

CHAPTER 14: Separating Letters from Words: Character Arrays versus Strings 235
Distinguishing Between a String and an Array of Characters 236
The String Container 236
Tapping into Your Library of String Functions 238
Gaining a New View of Strings 245
Taking a Deeper Dive into Formatting 246

Adding modifiers to your formatting 247
But wait! Why not just println(line)? 249
Changing Numbers to Strings 249

CHAPTER 15: **Pointing and Staring at Objects** 251
Declaring Arrays of Objects 251
Declaring Pointers to Objects 253
Dereferencing an object pointer 254
Pointing toward arrow pointers 255
Passing Objects to Functions 255
Calling a function with an object value 256
Calling a function with an object pointer 257
Calling a function by using the reference operator 258
Why Bother with Pointers or References? 260
Returning to the Heap 261
Allocating heaps of objects 262
When memory is allocated for you 263
Linking Up with Linked Lists 264
Performing other operations on a linked list 266
Hooking up with a LinkedListData program 267
A difference between unique and shared pointers 270
Ray of Hope: A List of Containers Linked to the C++ Library 273

CHAPTER 16: **Protecting Members: Do Not Disturb** 275
Making Members Private 276
Why you need private members 276
Discovering how private members work 277
Making an Argument for Using Private Members 279
Protecting the internal state of the class 279
Using a class with a limited interface 280
Getting and setting data members 280
Getting Friendly with Your Private Members 283

CHAPTER 17: **"Why Do You Build Me Up, Just to Tear Me Down, Baby?"** 287
Creating Objects 287
Using Constructors 289
Constructing a single object 289
Constructing multiple objects 291
Constructing a duplex 292
Dissecting a Destructor 294
Why you need the destructor 294
Working with destructors 294
Smart Pointers, Deleting, and Destruction — Oh, My! 298

CHAPTER 18: **Making Constructive Arguments** 301
Outfitting Constructors with Arguments 302
Placing Too Many Demands on the Carpenter: Overloading the Constructor 304
Sometimes, defaulting is good 306
Sometimes, using someone else's code is okay 307
Defaulting default Constructors 308
Constructing Class Members 310
Constructing a complex data member 310
Constructing a constant data member 316
Reconstructing the Order of Construction 317
Local objects construct in order 317
Static objects construct only once 317
All global objects construct before main() 319
Global objects construct in no particular order 319
Members construct in the order in which they are declared 320
Destructors destruct in the reverse order of the constructors 321
Constructing Arrays 321
Constructors as a Form of Conversion 322
Ignore That Value, Please 325

CHAPTER 19: **Making Copies with the Copy/Move Constructor** 327
Copying an Object 328
Why you need the copy constructor 329
Using the copy constructor 329
The Automatic Copy Constructor 331
Creating Shallow Copies versus Deep Copies 332
It's a Long Way to Temporaries 337
Avoiding temporaries permanently 339
Saying goodbye to copying 340
The Move Constructor 340
Just Because It Can Doesn't Mean It Will 343

CHAPTER 20: **Adding Static Members: Can Fabric Softener Help?** 345
Creating a Function that Remembers 345
Defining a Static Member 347
Why you need static members 347
Using static members 348
Referencing static data members 349
Uses for static data members 351
Keeping static variables inline 352

Declaring Static Member Functions 352
Static at a Global Level 355
What Is This About Anyway? 356

PART 4: A FIRST LOOK AT INHERITANCE 359

CHAPTER 21: Passing the DNA: Sharing Code with Inheritance 361
Do I Need My Inheritance? 362
How Does a Class Inherit? 364
Using a subclass 366
Let's talk about protecting your privates 366
Constructing a subclass 367
Destructing a subclass 368
Inheriting constructors 369
Having a HAS_A Relationship 369

CHAPTER 22: Creating Virtual Member Functions: Are They for Real? 371
Why You Need Polymorphism 374
How Polymorphism Works 375
When Is a Virtual Function Not? 376
Considering Virtual Considerations 378
Virtual Destructors 379
Let's Go on a Tangent and Reflect on a New Topic 380

CHAPTER 23: Factoring Classes 383
Factoring 384
Implementing Abstract Classes 388
Describing the abstract class concept 389
Making an honest class out of an abstract class 391
Passing abstract classes 391
Packing for Traveling (and Lunch) 392

PART 5: OBJECT-ORIENTED PROGRAMMING IN OVERDRIVE 395

CHAPTER 24: Adopting a New Assignment Operator, Should You Decide to Accept It 397
Comparing Operators with Functions 398
Inserting a New Operator 399
Creating Shallow Copies Is a Deep Problem 399
Overloading the Assignment Operator 401
No Copying Allowed! 405

Overloading the Subscript Operator 406
The Move Constructor and Move Operator 407

CHAPTER 25: **Playing with the Computer's File System** 413
Working with Files Using Filesystem 413
Creating and Working with Paths 414
Iterating over Folders 416
Does That File Exist? 417
Peeking at a File's Metadata 418
The File System and Files 419

CHAPTER 26: **Twice the Fun: Tapping into Multiple Inheritance** 421
Describing the Multiple Inheritance Mechanism 421
Straightening Out Inheritance Ambiguities 423
Adding Virtual Inheritance 424
Constructing the Objects of Multiple Inheritance 430
Interfacing with C++ 431
Is It Better to Have or to Be? That Is the Question 431
Reiterating a Contrary Opinion 432

CHAPTER 27: **Getting Ahead of Problems: Exception Handling, Contracts, and More** 435
Making an Exception to What I Just Said 436
Justifying a New Error Mechanism? 438
Examining the Exception Mechanism 439
What Kinds of Things Can You Throw? 442
Just Passing Through 445
Scratching the Surface of Classy Exceptions 446
Identifying What Is "Optional" 446
Were You Expected? 449
Using an Expected Value 451
Preventing Problems before They Can Happen: Contracts 452

PART 6: THE PART OF TENS 455

CHAPTER 28: **Ten Ways to Avoid Adding Bugs to Your Program** 457
Enable All Warnings and Error Messages 458
Adopt a Clear and Consistent Coding Style 458
Limit the Visibility 459
Comment Your Code While You Write It 461
Single-Step Every Path at Least Once 461
Avoid Overloading Operators 462

Manage the Heap Systematically 462
Use Exceptions to Handle Errors 463
Out with the New and in with the Newer 463
Avoid Multiple Inheritance 464

CHAPTER 29: **Ten Ways to Make Your Programs Easier to Update and Understand** 465
Pick Names Wisely 466
Use Comments 466
Write Modular Code 468
Steer Clear of Reinventing the Wheel 468
Style Your Code 469
Don't Leave Problems for Others 469
Test, Test, and Then Test Again 470
Use Modern Features 471
Avoid Hard-Coded Values 471
Use Version Control 472

INDEX 473