Contents at a Glance

Introduction 1

Book 1: Getting Started with C++ 7

CHAPTER 1: Configuring Your Desktop System. 9
CHAPTER 2: Configuring Your Mobile System 27
CHAPTER 3: Creating Your First C++ Application. 45
CHAPTER 4: Storing Data in C++. 69
CHAPTER 5: Directing the Application Flow 105
CHAPTER 6: Dividing Your Work with Functions 139
CHAPTER 7: Splitting Up Source Code Files 169
CHAPTER 8: Referring to Your Data Through Pointers 187

Book 2: Understanding Objects and Classes 225

CHAPTER 1: Working with Classes 227
CHAPTER 2: Using Advanced C++ Features 269
CHAPTER 3: Planning and Building Objects 309
CHAPTER 4: Building with Design Patterns 335

Book 3: Understanding Functional Programming 367

CHAPTER 1: Considering Functional Programming 369
CHAPTER 2: Working with Lambda Expressions 397
CHAPTER 3: Advanced Lambda Expressions 415

Book 4: Fixing Problems 427

CHAPTER 1: Dealing with Bugs 429
CHAPTER 2: Debugging an Application 443
CHAPTER 3: Stopping and Inspecting Your Code 457
CHAPTER 4: Traveling About the Stack 469

Book 5: Advanced Programming 479

CHAPTER 1: Working with Arrays, Pointers, and References 481
CHAPTER 2: Creating Data Structures 515
CHAPTER 3: Constructors, Destructors, and Exceptions 541
CHAPTER 4: Advanced Class Usage 571
CHAPTER 5: Creating Classes with Templates 601
CHAPTER 6: Programming with the Standard Library 637

COPYRIGHTED MATERIAL

Book 6: Reading and Writing Files 681
CHAPTER 1: Filing Information with the Streams Library 683
CHAPTER 2: Writing with Output Streams 697
CHAPTER 3: Reading with Input Streams 711
CHAPTER 4: Building Directories and Contents. 727
CHAPTER 5: Streaming Your Own Classes 737

Book 7: Advanced Standard Library Usage 751
CHAPTER 1: Exploring the Standard Library Further 753
CHAPTER 2: Working with User-Defined Literals (UDLs) 779
CHAPTER 3: Building Original Templates 795
CHAPTER 4: Investigating Boost. 817
CHAPTER 5: Boosting up a Step 849

Index 869

Table of Contents

INTRODUCTION 1
About This Book 1
Foolish Assumptions 4
Icons Used in This Book 4
Beyond the Book 5
Where to Go from Here 6

BOOK 1: GETTING STARTED WITH C++ 7

CHAPTER 1: Configuring Your Desktop System 9
Obtaining a Copy of C++ 20 10
Obtaining Code::Blocks 11
Installing Code::Blocks 12
Working with Windows 12
Working with Mac OS X 13
Using the standard Linux installation 14
Using the graphical Linux installation 15
Touring the Essential Code::Blocks Features 17
Starting Code::Blocks for the first time 18
Opening the sample projects 19
Viewing the essential windows 20
Using Other IDEs 25

CHAPTER 2: Configuring Your Mobile System 27
Obtaining CppDroid 28
Understanding why CppDroid is such a great choice 29
Getting your copy of CppDroid 31
Ensuring you get a good install 32
Considering Other Alternatives 32
Working with C4Droid 33
Getting multiple language support with AIDE 33
Using web-based IDEs 34
Touring the Essential CppDroid Features 35
Getting started with CppDroid 35
Accessing an example 37
Working with a simple online project 37
Accessing your source code 38
Considering differences with the desktop environment 39

Obtaining CppDroid Help 40
Working with the Help documentation 40
Getting community support 41
Using the free examples 42
Accessing the tutorials 43

CHAPTER 3: **Creating Your First C++ Application** 45
Code::Blocks Creating a Project 46
Understanding projects 46
Defining your first project 47
Building and executing your first application 52
Typing the Code 53
Starting with Main 55
Showing Information 55
Doing some math 60
Tabbing your output 66
Let Your Application Run Away 67

CHAPTER 4: **Storing Data in C++** 69
Putting Your Data Places: Variables 70
Creating an integer variable 70
Declaring multiple variables 73
Changing values 74
Setting one variable equal to another 74
Initializing a variable 75
Creating a great name for yourself 76
Manipulating Integer Variables 78
Adding integer variables 78
Subtracting integer variables 82
Multiplying integer variables 84
Dividing integer variables 86
Characters 88
Null character 89
Nonprintable and other cool characters 89
Strings 93
Getting a part of a string 94
Changing part of a string 95
Adding onto a string 96
Adding two strings 97
Making Decisions Using Conditional Operators 98
Telling the Truth with Boolean Variables 100
Reading from the Console 102

CHAPTER 5: **Directing the Application Flow** 105
Doing This or Doing That 106
Evaluating Conditions in C++ 107
Finding the right C++ operators 108
Combining multiple evaluations 110
Including Evaluations in C++ Conditional Statements 111
Deciding what if and also what else 112
Going further with the else and if 113
Repeating Actions with Statements That Loop 115
Understanding how computers use loops 116
Looping situations 116
Looping for 117
Performing a simple for loop 118
Using multiple initialization variables 123
Working with ranges 126
Placing a condition within the declaration 128
Letting C++ determine the type 129
Looping while 130
Doing while 132
Breaking and continuing 133
Breaking 134
Continuing 135
Nesting loops 136

CHAPTER 6: **Dividing Your Work with Functions** 139
Dividing Your Work 139
Calling a Function 144
Passing a variable 146
Passing multiple variables 147
Writing Your Own Functions 148
Defining the AddOne() function 149
Seeing how AddOne() is called 150
Taking the AddOne() Function apart 150
Considering the AddOne() parameter 151
Understanding the AddOne() name and type 152
Improving On the Basic Function 153
Using multiple parameters or no parameters 153
Returning nothing 156
Keeping your variables local 157
Forward references and function prototypes 159
Writing two versions of the same function 161

Calling All String Functions 163
Inserting a string into a string 163
Removing parts of a string 164
Replacing parts of a string 164
Using the string functions together 164
Understanding main() 165

CHAPTER 7: **Splitting Up Source Code Files** 169
Creating Multiple Source Files 170
Adding a new source code file 170
Removing an existing source code file 173
Creating a project with multiple existing files 173
Getting multiple files to interact 177
Sharing with Header Files 179
Adding the header only once 182
Using angle brackets or quotes 182
Sharing Variables among Source Files 183
Using the Mysterious Header Wrappers 185

CHAPTER 8: **Referring to Your Data Through Pointers** 187
Understanding the Changes in Pointers for C++ 20 188
Avoiding broken code 188
Considering the issues 189
Writing cleaner and less bug-prone code 191
Heaping and Stacking the Variables 192
Getting a variable's address 196
Changing a variable by using a pointer 198
Pointing at a string 200
Pointing to something else 203
Tips on pointer variables 204
Creating New Raw Pointers 205
Using new 206
Using an initializer 208
Freeing Raw Pointers 209
Working with Smart Pointers 211
Creating smart pointers using std::unique_ptr and std::shared_ptr 212
Defining nullable values using std::optional and std::nullopt 216
Passing Pointer Variables to Functions 218
Returning Pointer Variables from Functions 221

PART 2: UNDERSTANDING OBJECTS AND CLASSES 225

CHAPTER 1: Working with Classes 227

Understanding Objects and Classes 227
Classifying classes and objects 230
Describing methods and data 231
Implementing a class 232
Separating method code 237
The parts of a class 240
Working with a Class 241
Accessing members 241
Using classes and raw pointers 244
Using classes and smart pointers 248
Passing objects to functions 249
Using const parameters in functions 251
Using the this pointer 252
Overloading methods 256
Starting and Ending with Constructors and Destructors 259
Starting with constructors 259
Ending with destructors 260
Sampling constructors and destructors 260
Adding parameters to constructors 263
Building Hierarchies of Classes 264
Creating a hierarchy in C++ 265
Understanding types of inheritance 266
Creating and Using Object Aliases 267

CHAPTER 2: Using Advanced C++ Features 269

Filling Your Code with Comments 270
Converting Types 272
Understanding how int and string conversions work 272
Seeing int and string conversions in action 273
Considering other conversion issues 276
Reading from the Console 277
Understanding Preprocessor Directives 282
Understanding the basics of preprocessing 282
Creating constants and macros with #define 283
Performing conditional compilation 286
Exercising the basic preprocessor directives 288
Using Constants 292
Using Switch Statements 295

Supercharging enums with Classes 298
Working with Random Numbers 300
Storing Data in Arrays 302
Declaring and accessing an array 303
Arrays of pointers 304
Passing arrays to functions 306
Adding and subtracting pointers 307

CHAPTER 3: **Planning and Building Objects** 309
Recognizing Objects 310
Observing the Mailboxes class 312
Observing the Mailbox class 314
Finding other objects 315
Encapsulating Objects 316
Considering the Application Programming Interface 316
Understanding properties 316
Choosing between private and protected 318
Defining a process 318
Implementing properties 319
Building Hierarchies 322
Establishing a hierarchy 322
Protecting members when inheriting 324
Overriding methods 330
Specializing with polymorphism 332
Getting abstract about things 333

CHAPTER 4: **Building with Design Patterns** 335
Delving Into Pattern History 336
Introducing a Simple Pattern: the Singleton 337
Using an existing pattern 337
Creating a singleton pattern class 338
Watching an Instance with an Observer 341
Understanding the observer pattern 341
Defining an observer pattern class 343
Observers and the Standard C++ Library 346
Automatically adding an observer 347
Mediating with a Pattern 349
Defining the mediator pattern scenario 350
Outlining the car example 351
Creating the car example 354

BOOK 3: UNDERSTANDING FUNCTIONAL PROGRAMMING 367

CHAPTER 1: Considering Functional Programming 369
Understanding How Functional Programming Differs 370
Defining an Impure Language 373
Considering the requirements 373
Understanding the C++ functional limitations 374
Seeing Data as Immutable 375
Working with immutable variables 376
Working with immutability in classes and structures 377
Creating constant expressions 378
Considering the Effects of State 381
Eliminating Side Effects 382
Contrasting declarations and functions 383
Associating functions with side effects 384
Removing side effects 385
Creating a declarative C++ example 387
Understanding the Role of auto 388
Passing Functions to Functions 390
Seeing a simple example of function input 391
Using transforms 393
Using Lambda Expressions for Implementation 394

CHAPTER 2: Working with Lambda Expressions 397
Creating More Readable and Concise C++ Code 398
Defining the Essential Lambda Expression 399
Defining the parts of a lambda expression 399
Relying on computer detection of return type 401
Using the auto keyword with lambda expressions 404
Using lambda expressions as macros 405
Developing with Lambda Expressions 406
Using lambda expressions with classes and structures 407
Working with the capture clause 408
Sorting data using a lambda expression 411
Specifying that the lambda expression throws exceptions 413

CHAPTER 3: Advanced Lambda Expressions 415
Considering the C++ 20 Lambda Extensions 416
Defining an immediate function 416
Using = and this in captures 417
Finding other changes 418

Working in Unevaluated Contexts 418
Using Assignable Stateless Lambda Expressions 420
Dealing with Pack Expansions 422
Considering the template 422
Processing the variables using recursion 423
Processing the variables using a lambda expression 424

BOOK 4: FIXING PROBLEMS 427

CHAPTER 1: Dealing with Bugs 429
It's Not a Bug. It's a Feature! 430
Make Your Application Features Look Like Features 431
Anticipating (Almost) Everything 432
Considering menus 432
Dealing with textual input 435
Performing string processing 437
Avoiding Mistakes, Plain and Simple 441

CHAPTER 2: Debugging an Application 443
Programming with Debuggers 444
Running the debugger 446
Recognizing the parts of the Code::Blocks debugger 453
Debugging with Different Tools 455
Debugging a Code::Blocks Application with Command-Line Arguments 456

CHAPTER 3: Stopping and Inspecting Your Code 457
Setting and Disabling Breakpoints 458
Setting a breakpoint in Code::Blocks 459
Enabling and disabling breakpoints 460
Watching, Inspecting, and Changing Variables 463
Watching the variables 465
Changing values 466

CHAPTER 4: Traveling About the Stack 469
Stacking Your Data 470
Moving about the stack 471
Storing local variables 473
Debugging with Advanced Features 475
Viewing threads 475
Tracing through assembly code 475

BOOK 5: ADVANCED PROGRAMMING 479

CHAPTER 1: Working with Arrays, Pointers, and References 481

Building Up Arrays 482
Declaring arrays 482
Arrays and pointers 484
Using multidimensional arrays 488
Arrays and command-line parameters 492
Allocating an array on the heap 494
Deleting an array from the heap 494
Storing arrays of pointers and arrays of arrays 495
Building constant arrays 498
Pointing with Pointers 498
Becoming horribly complex 499
Pointers to functions 505
Pointing a variable to a method 506
Pointing to static methods 509
Referring to References 510
Reference variables 510
Returning a reference from a function 511

CHAPTER 2: Creating Data Structures 515

Working with Data 515
The great variable roundup 516
Expressing variables from either side 518
Casting a spell on your data 520
Comparing casting and converting 521
Casting safely with C++ 523
Structuring Your Data 529
Structures as component data types 531
Equating structures 531
Returning compound data types 532
Naming Your Space 534
Creating a namespace 534
Employing using namespace 535
Using variables 537
Using part of a namespace 538

CHAPTER 3: Constructors, Destructors, and Exceptions 541

Constructing and Destructing Objects 542
Overloading constructors 542
Initializing members 543
Adding a default constructor 548
Functional constructors 550

Calling one constructor from another 553
Copying instances with copy constructors 555
When constructors go bad. 557
Destroying your instances 558
Virtually inheriting destructors 560
Programming the Exceptions to the Rule 563
Creating a basic try. . .catch block. 563
Using multiple catch blocks 565
Throwing direct instances 566
Catching any exception. 567
Rethrowing an exception 568
Using a standard category 570

CHAPTER 4: **Advanced Class Usage** 571
Inherently Inheriting Correctly 572
Morphing your inheritance 572
Avoiding polymorphism 573
Adjusting access. 574
Avoiding variable naming conflicts 575
Using class-based access adjustment 576
Returning something different, virtually speaking 577
Multiple inheritance 581
Virtual inheritance. 584
Friend classes and functions 588
Using Classes and Types within Classes. 591
Nesting a class 591
Types within classes 597

CHAPTER 5: **Creating Classes with Templates** 601
Templatizing a Class 602
Considering types 602
Defining the need for templates 602
Creating and using a template 605
Understanding the template keyword 607
Going Beyond the Basics 609
Separating a template from the function code. 609
Including static members in a template. 611
Parameterizing a Template 612
Putting different types in the parameter 613
Including multiple parameters 616
Working with non-type parameters 619

Typedefing a Template 622
Deriving Templates 623
Deriving classes from a class template 623
Deriving a class template from a class 626
Deriving a class template from a class template 627
Templatizing a Function 630
Overloading and function templates 632
Templatizing a method 635

CHAPTER 6: **Programming with the Standard Library** 637
Architecting the Standard Library 638
Containing Your Classes 638
Storing in a vector 639
Working with std::array 642
Mapping your data 643
Containing instances, pointers, or references 644
Working with copies 648
Comparing instances 649
Iterating through a container 655
A map of pairs in your hand 658
The Great Container Showdown 658
Associating and storing with a set 658
Unionizing and intersecting sets 662
Listing with list 664
Stacking the deque 669
Waiting in line with stacks and queues 670
Copying Containers 673
Creating and Using Dynamic Arrays 675
Working with Unordered Data 677
Using std::unordered_set to create an unordered set 677
Manipulating unordered sets 677
Working with Ranges 679

BOOK 6: READING AND WRITING FILES 681

CHAPTER 1: **Filing Information with the Streams Library** 683
Seeing a Need for Streams 684
Programming with the Streams Library 686
Getting the right header file 686
Opening a file 687
Reading from a file 690
Reading and writing a file 691
Working with containers 692
Handling Errors When Opening a File 693
Flagging the ios Flags 695

CHAPTER 2: **Writing with Output Streams** 697
Inserting with the << Operator 698
Formatting Your Output 699
Formatting with flags 700
Specifying a precision 704
Setting the width and creating fields 707

CHAPTER 3: **Reading with Input Streams** 711
Extracting with Operators 712
Encountering the End of File 715
Using the record count approach 715
Using the EOF check approach 718
Reading Various Types 720
Understanding data reading issues 720
Writing and reading string-type data 721
Writing and reading structured data 724

CHAPTER 4: **Building Directories and Contents** 727
Manipulating Directories 728
Creating a directory 728
Deleting a directory 730
Getting the Contents of a Directory 731
Copying Files 733
Copying with windows 734
Using the quick-and-dirty method 734
Moving and Renaming Files and Directories 735

CHAPTER 5: **Streaming Your Own Classes** 737
Streaming a Class for Text Formatting 738
Understanding the process 739
Considering the insertion implementation 739
Considering the extraction implementation 741
Manipulating a Stream 742
What's a manipulator? 742
Writing your own manipulator 744

BOOK 7: ADVANCED STANDARD LIBRARY USAGE 751

CHAPTER 1: **Exploring the Standard Library Further** 753
Considering the Standard Library Categories 755
Algorithms 755
Atomic operations 757
C Compatibility 759
Concepts 759

Containers 760
Coroutines 760
Filesystem 761
Input/Output 761
Iterators 761
Localization 763
Numerics 763
Ranges 764
Regular Expressions 766
Strings 766
Thread Support 767
Utilities 767
Parsing Strings Using a Hash 768
Obtaining Information Using a Random Access Iterator 771
Locating Values Using the Find Algorithm 774
Using the Random Number Generator 776
Working with Temporary Buffers 777

CHAPTER 2: **Working with User-Defined Literals (UDLs)** 779
Understanding the Need for UDLs 780
Prefixes and suffixes 781
Differentiating between raw and cooked 784
Working with the UDLs in the Standard Library 785
std::basic_string 785
std::complex 788
std::chrono::duration 789
Creating Your Own UDLs 791
Developing a conversion UDL 792
Developing a custom type UDL 793
Using a custom UDL for side effects 794

CHAPTER 3: **Building Original Templates** 795
Deciding When to Create a Template 796
Defining the Elements of a Good Template 797
Creating a Basic Math Template 799
Building a Structure Template 801
Developing a Class Template 804
Considering Template Specialization 807
Creating a Template Library 809
Defining the library project 810
Configuring the library project 812
Coding the library 813
Using Your Template Library 815

CHAPTER 4: **Investigating Boost** 817
Considering the Standard Library Alternative 818
Understanding why the Standard Library contains Boost features 818
Defining the trade-offs of using the Standard Library 819
Understanding Boost 820
Boost features 821
Licensing 822
Paid support 823
Obtaining and Installing Boost for Code::Blocks 823
Unpacking Boost 823
Using the header-only libraries 825
Building the libraries 825
Testing the installation 827
Creating the Boost Tools 833
Using Boost.Build 836
Getting a successful build 836
Creating your own example 836
Using Inspect 837
Understanding BoostBook 840
Using QuickBook 841
Using bcp 843
Using Wave 845
Building Your First Boost Application Using Date Time 846

CHAPTER 5: **Boosting up a Step** 849
Parsing Strings Using RegEx 850
Adding the RegEx library 851
Creating the RegEx code 855
Breaking Strings into Tokens Using Tokenizer 857
Performing Numeric Conversion 858
Creating Improved Loops Using Foreach 862
Accessing the Operating System Using Filesystem 864

INDEX 869