

LEARNING MADE EASY



4th Edition

# C++

ALL-IN-ONE

for  
**dummies**<sup>®</sup>  
A Wiley Brand



**John Paul Mueller**

Author of *Functional Programming  
For Dummies*





# C++

## ALL-IN-ONE

4th Edition

**by John Paul Mueller**

for  
**dummies**<sup>®</sup>  
A Wiley Brand

## C++ All-in-One For Dummies®, 4th Edition

Published by: **John Wiley & Sons, Inc.**, 111 River Street, Hoboken, NJ 07030-5774, [www.wiley.com](http://www.wiley.com)

Copyright © 2021 by John Wiley & Sons, Inc., Hoboken, New Jersey

Published simultaneously in Canada

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without the prior written permission of the Publisher. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

**Trademarks:** Wiley, For Dummies, the Dummies Man logo, Dummies.com, Making Everything Easier, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and may not be used without written permission. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc. is not associated with any product or vendor mentioned in this book.

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: THE PUBLISHER AND THE AUTHOR MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES OR PROMOTIONAL MATERIALS. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR EVERY SITUATION. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING LEGAL, ACCOUNTING, OR OTHER PROFESSIONAL SERVICES. IF PROFESSIONAL ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL PERSON SHOULD BE SOUGHT. NEITHER THE PUBLISHER NOR THE AUTHOR SHALL BE LIABLE FOR DAMAGES ARISING HEREFROM. THE FACT THAT AN ORGANIZATION OR WEBSITE IS REFERRED TO IN THIS WORK AS A CITATION AND/OR A POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE AUTHOR OR THE PUBLISHER ENDORSES THE INFORMATION THE ORGANIZATION OR WEBSITE MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. FURTHER, READERS SHOULD BE AWARE THAT INTERNET WEBSITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ.

For general information on our other products and services, please contact our Customer Care Department within the U.S. at 877-762-2974, outside the U.S. at 317-572-3993, or fax 317-572-4002. For technical support, please visit <https://hub.wiley.com/community/support/dummies>.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at <http://booksupport.wiley.com>. For more information about Wiley products, visit [www.wiley.com](http://www.wiley.com).

Library of Congress Control Number: 2020949804

ISBN: 978-1-119-60174-6

ISBN 978-1-119-60175-3 (ebk); ISBN 978-1-119-60173-9 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

# Contents at a Glance

<b>Introduction</b>	1
<b>Book 1: Getting Started with C++</b>	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
<b>Book 2: Understanding Objects and Classes</b>	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
<b>Book 3: Understanding Functional Programming</b>	367
CHAPTER 1: Considering Functional Programming	369
CHAPTER 2: Working with Lambda Expressions	397
CHAPTER 3: Advanced Lambda Expressions	415
<b>Book 4: Fixing Problems</b>	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
<b>Book 5: Advanced Programming</b>	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

<b>Book 6: Reading and Writing Files</b> .....	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
 <b>Book 7: Advanced Standard Library Usage</b> .....	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
 <b>Index</b> .....	869

# Table of Contents

<b>INTRODUCTION</b>	1
About This Book	1
Foolish Assumptions	4
Icons Used in This Book	4
Beyond the Book	5
Where to Go from Here	6
<b>BOOK 1: GETTING STARTED WITH C++</b>	7
<b>CHAPTER 1: Configuring Your Desktop System</b>	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
<b>CHAPTER 2: Configuring Your Mobile System</b>	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
<b>CHAPTER 3: Creating Your First C++ Application .....</b>	<b>45</b>
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
<b>CHAPTER 4: Storing Data in C++ .....</b>	<b>69</b>
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



<b>CHAPTER 5: Directing the Application Flow</b>	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
<b>CHAPTER 6: Dividing Your Work with Functions</b>	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
<b>CHAPTER 7: Splitting Up Source Code Files. . . . .</b>	<b>169</b>
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
<b>CHAPTER 8: Referring to Your Data Through Pointers . . . . .</b>	<b>187</b>
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 <code>std::unique_ptr</code> and <code>std::shared_ptr</code> . . . . .	212
Defining nullable values using <code>std::optional</code> and <code>std::nullopt</code> . . . . .	216
Passing Pointer Variables to Functions . . . . .	218
Returning Pointer Variables from Functions. . . . .	221

<b>PART 2: UNDERSTANDING OBJECTS AND CLASSES</b>	225
<b>CHAPTER 1: Working with Classes</b>	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
<b>CHAPTER 2: Using Advanced C++ Features</b>	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
<b>CHAPTER 3: Planning and Building Objects .....</b>	<b>309</b>
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
<b>CHAPTER 4: Building with Design Patterns.....</b>	<b>335</b>
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

<b>BOOK 3: UNDERSTANDING FUNCTIONAL PROGRAMMING</b>	<b>367</b>
<b>CHAPTER 1: Considering Functional Programming</b>	<b>369</b>
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
<b>CHAPTER 2: Working with Lambda Expressions</b>	<b>397</b>
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
<b>CHAPTER 3: Advanced Lambda Expressions</b>	<b>415</b>
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
<b>BOOK 4: FIXING PROBLEMS . . . . .</b>	<b>427</b>
<b>CHAPTER 1: Dealing with Bugs . . . . .</b>	<b>429</b>
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
<b>CHAPTER 2: Debugging an Application . . . . .</b>	<b>443</b>
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
<b>CHAPTER 3: Stopping and Inspecting Your Code . . . . .</b>	<b>457</b>
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
<b>CHAPTER 4: Traveling About the Stack . . . . .</b>	<b>469</b>
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

<b>CHAPTER 1:</b>	<b>Working with Arrays, Pointers, and References.....</b>	<b>481</b>
	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
<b>CHAPTER 2:</b>	<b>Creating Data Structures.....</b>	<b>515</b>
	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
<b>CHAPTER 3:</b>	<b>Constructors, Destructors, and Exceptions.....</b>	<b>541</b>
	Constructing and Destructing Objects.....	542
	Overloading constructors.....	542
	Initializing members.....	543
	Adding a default constructor.....	548
	Functional constructors.....	550