LEARNING MADE EASY

4th Edition

C++

ALL-IN-ONE

for dummies®
A Wiley Brand

7 Books in one!

John Paul Mueller
Author of *Functional Programming For Dummies*

C++

ALL-IN-ONE

4th Edition

by John Paul Mueller

for dummies®
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

Published simultaneously in Canada

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

Table of Contents

Introduction

There are many general-purpose programming languages today, but few can claim to be the language of the millennium. C++ can make that claim, and for good reason:

- It's powerful. You can write almost any program in it.
- It's fast, and it's fully compiled**.** That's a good thing.
- It's easy to use — if you have this book.
- It's object oriented. If you're not sure what that is, don't worry. You can find out about it by reading this very book you're holding.
- It supports functional programming techniques, which makes modeling math problems considerably easier and makes parallel processing easier. This book covers functional programming techniques, too.
- It's portable. Versions are available for nearly every computer.
- It's standardized. The American National Standards Institute (ANSI) and the International Standards Organization (ISO) both approve an official version.
- It's continually updated to meet the changing challenges of the computer community.
- It's popular. More people are using C++ because so many other people use it.

Sure, some people criticize C++. But most of these people don't truly understand C++ or are just having a bad day. Or both.

About This Book

This book is a hands-on, roll-up-your-sleeves experience that gives you the opportunity to truly learn C++. This edition starts out by helping you get a great C++ installation in place. A lot of readers wrote to tell me that they simply couldn't get C++ to work for them, and I listened by adding configuration instructions in Book 1, Chapter 1. You can find instructions for working with the Mac, Linux, and Windows throughout the book. The examples are also tested to work on all three platforms.

C++ All-in-One For Dummies, 4th Edition, is devoted to working with C++ wherever you want to use it. Book 1, Chapter 2 even includes techniques for writing C++ code on your mobile device, although writing a complex application on your smartphone would be understandably difficult because of the small device size.

At the very beginning, I start you out from square one. I don't assume any programming experience whatsoever. Everybody has to start somewhere. You can start here. Not to brag, but you are in the hands of a highly successful C++ developer who has shown thousands of people how to program, many of whom also started out from square one.

You already know C++? This book is great for you, too, because although I start discussing C++ from the beginning, I cover the important aspects of the language in depth. Even if you've used C++ in the past, this book gets you up to speed with the latest in C++ 14 and above innovations, including C++ 20 additions. Plus, this edition of the book focuses on all the latest programming strategies while removing some of the less used functionality of the past.

If you're interested in using the time-tested Object Oriented Programming (OOP) techniques that C++ developers have used for years, then Book 2 is where you want to look. You start with a view of classes, but eventually move into more advanced topics, including the use of programming patterns in Book 2 Chapter 4.

One of the most exciting additions to this edition is the use of functional programming techniques, which you can find in Book 3. Functional programming has become extremely popular because it makes modeling math problems significantly easier, and many people use functional programming techniques to solve modern data science problems. More important, functional programming can be a lot easier than earlier programming paradigms.

Every application out there has a bug or two. If you doubt this statement, just try to find one that is bug free—you won't. Book 4 includes all sorts of techniques you can use to make your application as bug free as possible before it leaves your machine and then help you find the bugs that others graciously point out later.

Book 5 is all about moving you from generalized programming strategies into the advanced strategies used by modern developers. It starts with a look at standardized structures for working with classes in a safe manner. The minibook takes you through

- Simple structures, such as arrays
- More advanced data management
- The use of constructors, destructors, and exceptions

- Templatized programming
- Use of the Standard Library (originally called the Standard Template Library or STL).

Everyone needs to work with files at some point. You use local, network, and Internet files today on a regular basis. Book 6 is all about working with files in various ways. This book includes topics on working with data streams as well.

The Standard Library is immense and there are entire books written about its use. *C++ All-in-One For Dummies*, 4th Edition, focuses on providing you with a really good overview that you can use to drill down into more detailed topics later. Besides looking at the Standard Library in more detail, you discover how to work with User Defined Literals (UDLs) and how to create your own templates. This book also delves into the Boost library, which is the library that has added more to Standard Library than just about any other source. Check out Book 7, Chapters 4 and 5 to learn about Boost. If you use C++ and don't use Boost, you're really missing out!

C++ is standardized, and you can use the information in this book on many different platforms. I wrote the samples using Mac OS X, SUSE Linux (some of the beta readers used other flavors of Linux), and Windows systems (with some testing on my ASUS tablet as well). In order to make this happen, I used a compiler called *Code::Blocks* that runs on almost every computer (Windows, Linux, and Macintosh) and CppDroid for my tablet. It doesn't matter which device you're using!

To make absorbing the concepts easy, this book uses the following conventions:

- Text that you're meant to type just as it appears in the book is in **bold**. The exception is when you're working through a step list: Because each step is bold, the text to type is not bold.
- Web addresses and programming code appear in `monofont`. If you're reading a digital version of this book on a device connected to the Internet, you can click or tap the web address to visit that website, like this: `https://www.dummies.com`.
- When you need to type command sequences, you see them separated by a special arrow, like this: File➪New File. In this example, you go to the File menu first and then select the New File entry on that menu.
- When you see words in *italics* as part of a typing sequence, you need to replace that value with something that works for you. For example, if you see "Type ***Your Name*** and press Enter," you need to replace *Your Name* with your actual name.

Foolish Assumptions

This book is designed for novice and professional alike. You can either read this book from cover to cover, or you can look up topics and treat the book as a reference guide — whichever works best for you. Keep it on your shelf, and have it ready to grab when you need to look something up. However, I've made some assumptions about your level of knowledge when I put the book together. The most important of these assumptions is that you already know how to use your device and work with the operating system that supports it. You also need to know how to perform tasks like downloading files and installing applications. A familiarity with the Internet is also required, and you need to know how to interact with it moderately well to locate the resources you need to work with the book. Finally, you must know how to work with archives, such as the ZIP file format.

Icons Used in This Book

As you read this book, you see icons in the margins that indicate material of interest (or not, as the case may be). This section briefly describes each icon in this book.


TIP

Tips are nice because they help you save time or perform some task without a lot of extra work. The tips in this book are time-saving techniques or pointers to resources that you should try so that you can get the maximum benefit from C++. Most important, many of these tips will help you make sense of the overwhelming quantity of libraries and tools that C++ developers have created over the years.


WARNING

I don't want to sound like an angry parent or some kind of maniac, but you should avoid doing anything that's marked with a Warning icon. Otherwise, you might find that your application fails to work as expected, you get incorrect answers from seemingly bulletproof code, or (in the worst-case scenario) you lose data. Given where C++ appears, you might also send the next rocket off to Mars prematurely, make someone's thermostat misbehave, or cause nationwide power outages. Really, warnings are for everyone!


TECHNICAL STUFF

Whenever you see this icon, think advanced tip or technique. You might find these tidbits of useful information just too boring for words, or they could contain the solution you need to get a program running. Skip these bits of information whenever you like.

If you don't get anything else out of a particular chapter or section, remember the material marked by this icon. This text usually contains an essential process or a bit of information that you must know to work with C++, or to perform development tasks successfully.

Beyond the Book

If you want to email me, please do! Make sure you send your book-specific requests to:

```
John@JohnMuellerBooks.com
```

I get a lot of email from readers, so sometimes it takes me a while to answer. I try very hard to answer every book-specific question I receive, though, so I highly recommend contacting me with your questions. I want to ensure that your book experience is the best one possible. The blog category at `http://blog.johnmuellerbooks.com/categories/263/c-all-in-one-for-dummies.aspx` contains a wealth of additional information about this book. You can check out the website at `http://www.johnmuellerbooks.com/`.

This book isn't the end of your C++ programming experience — it's really just the beginning. I provide online content to make this book more flexible and better able to meet your needs. That way, as I receive email from you, I can address questions and tell you how updates to either Code::Blocks or the C++ language affect book content. You can also access other cool materials:

- **Cheat Sheet:** You remember using crib notes in school to make a better mark on a test, don't you? You do? Well, a cheat sheet is sort of like that. It provides you with some special notes on things you can do with C++ that not every other developer knows. You can find the cheat sheet for this book at `www.dummies.com` and typing **C++ All-in-One For Dummies, 4th Edition** in the search field. It contains really neat information like the top ten mistakes developers make when working with C++, a list of header files that you use in most applications, and some of the C++ syntax that gives most developers problems.
- **Updates:** Sometimes changes happen. For example, I might not have seen an upcoming change when I looked into my crystal ball during the writing of this book. In the past, such a situation simply meant that the book would become outdated and less useful, but you can now find updates to the book at `www.dummies.com`. In addition to these updates, check out the blog posts with

answers to reader questions and demonstrations of useful book-related techniques at http://blog.johnmuellerbooks.com/.

- **Companion files:** Hey! Who really wants to type all the code in the book? Most readers would prefer to spend their time actually working through coding examples rather than typing. Fortunately for you, the source code is available for download, so all you need to do is read the book to learn C++ coding techniques. Each of the book examples even tells you precisely which example project to use. You can find these files by visiting www.dummies.com/go/caiofd4e.

  Just in case you're worried about Code::Blocks, you can find complete download and installation instructions for it in Book 1, Chapter 1. Don't worry about which platform you use. This chapter includes instructions for Mac OS X, Linux, and Windows.

Where to Go from Here

If you're just starting your C++ adventure, I highly recommend starting at either Book 1, Chapter 1 (for desktop developers) or Book 1, Chapter 2 (for mobile developers). You really do need to create a solid foundation before you can tackle the code in this book. If you're in a hurry and already have a C++ installation, you can always try starting with Book 1, Chapter 3.

Readers with a little more experience, who already know some C++ basics, can skip some of these introductory chapters, but you definitely don't want to skip Book 1, Chapter 8 because it contains a lot of pointer-related changes in current versions of C++. If you skip this chapter, you may find later that you have a hard time following the example code in the book because the newer examples use these pointer features.

An advanced reader with some idea of the current changes in C++ 20 could possibly skip Book 1, but scanning Book 2 is a good idea because there are some OOP changes you definitely want to know about. However, even for advanced readers, skipping Book 3 is a bad idea because modern development really is moving toward functional programming techniques.

1 Getting Started with C++

Contents at a Glance

and more . . .

IN THIS CHAPTER

- Getting your own copy of C++ 20
- Getting a copy of Code::Blocks
- Creating a Code::Blocks work environment on your system
- Seeing how Code::Blocks helps you perform tasks
- Working with other IDEs

Chapter 1
Configuring Your Desktop System

This chapter is for those of you who have a desktop system and want to use it to create your application code. Chapter 2 discusses how to perform the same task using a mobile device (and provides you with some trade-offs between the two environments). Whether you use the desktop or the mobile solution, you need a copy of a compiler that supports C++ 20 features or some book examples won't work at all. This book relies on the GNU Compiler Collection (GCC) version 8.3 compiler because it provides great C++ 20 support (see `https://en.cppreference.com/w/cpp/compiler_support`). The best way to obtain the version 8.3 compiler for your desktop system is to follow the steps in this chapter.

Before you can do anything interesting at all with C++, you need a copy of it installed on your system. Of course, this means going online, finding the location of the software that's appropriate for your platform, and then downloading it as necessary. If you use an Integrated Development Environment (IDE) such as Code::Blocks (the IDE used throughout this book), you get a copy of C++ with your installation, so you don't need to worry about reading the first section of this chapter. This book relies on your having a compiler capable of compiling C++ 20 code, which is the latest version of the language available at the time of this writing.

REMEMBER

Even though this book focuses on working with C++ on the Mac, Windows, and Linux platforms, you can actually use the techniques it provides on a great many other desktop systems. With this in mind, you'll find an overview of using C++ with other IDEs. As your platform becomes more esoteric, you'll find that fewer of the book examples work because your platform may require special programming techniques. The best option for working with this book is using a copy of Code::Blocks 17.12 with C++ 20 support installed on the Linux, Mac, or Windows platform.

Obtaining a Copy of C++ 20

There is no product available named C++ 20. The C++ 20 standard simply says what the language contains and how someone should implement it. In other words, you can't just go online and get a copy of C++ 20; what you need to do instead is get a compiler vendor's implementation of the C++ 20 standard. For example, you can download the GNU Compiler Collection (GCC) version of C++ 20 from `https://gcc.gnu.org/releases.html`.

TIP

Every vendor will have a slightly different interpretation of this standard and could provide additions to the standard. In short, every compiler provides a unique version of C++. However, you also have the choice of not using the special features that the vendor provides, which means your source code is less susceptible to problems that occur when you use multiple compilers. The examples in this book are strictly written to the C++ 20 standard, so you shouldn't have a problem using them anywhere you want.

It's important that you also understand that a compiler is not the same as an Integrated Development Environment (IDE). The compiler is separate from the IDE in many cases and maintained by two separate parties. For example, the Code::Blocks IDE supports multiple compilers, and the GCC compiler works within multiple IDEs. The compiler is the important piece of software that turns your source code into an executable file that the operating system can run to produce the output you want.

The compiler you choose has to support the platforms you want to work with. For example, GCC supports Mac, Windows, and Linux development as well as some Acorn or (later) Advanced RISC Machine (ARM) processors (ARM doesn't officially stand for anything today). In fact, it may support other platforms by the time you read this chapter. Because it works in so many places, this book focuses on GCC, even though the examples will work with other compilers with some modification to overcome compiler differences.