



O'REILLY



Preface

The C++ of today provides programmers with the ability to write expressive and robust code, while still making it possible to target almost any hardware platform, and at the same time meet performance-critical requirements. This makes C++ a unique language. Over the last few years, C++ has turned into a modern language that is more fun to use and has better defaults.

This book aims to give you a solid foundation to write efficient applications, as well as an insight into strategies for implementing libraries in modern C++. I have tried to take a practical approach to explaining how C++ works today, where features from C++17 and C++20 are a natural part of the language, rather than looking at C++ historically.

This second edition was written to cover new features added in C++20. I have included features that I think fit well with the rest of the content and the focus of this book. Naturally, chapters that discuss new features serve more as an introduction and contain fewer best practices and well-proven solutions.

At the time of publishing this book, the compiler support for some C++20 features presented is still experimental. If you read this book near to the publication date, the chances are that you will have to wait for some features to become fully supported by your compiler.

Many chapters span a wide range of difficulty. They start with the absolute basics and end with advanced topics such as custom memory allocators. If a section is not relevant to you, feel free to skip it or come back to it later. Apart from the first three chapters, most chapters can be read independently.

Our main technical reviewer, Timur Doumler, has had a big impact on this new edition. His enthusiasm and brilliant feedback have led to some chapters from the first edition being reworked to explain topics more thoroughly and in more depth. Timur has also been a vital contributor when it comes to incorporating new C++20 features into the chapters where they fit naturally. Select parts of the book have also been reviewed by Arthur O'Dwyer, Marius Bancila, and Lewis Baker. It has been a true pleasure to have had such excellent reviewers on this project. I hope you enjoy reading this new edition as much as I have enjoyed writing it.

Who this book is for

This book expects you to have a basic knowledge of C++ and computer architecture, and a genuine interest in evolving your skills. Hopefully, by the time you finish this book, you will have gained a few insights into how you can improve your C++ applications, both performance-wise and syntactically. On top of that, I also hope that you will have a few "aha" moments.

What this book covers

Chapter 1, A Brief Introduction to C++, introduces some important properties of C++, such as zero-cost abstractions, value semantics, const correctness, explicit ownership, and error handling. It also discusses the drawbacks of C++.

Chapter 2, Essential C++ Techniques, outlines automatic type deduction using `auto`, lambda functions, move semantics, and error handling.

Chapter 3, Analyzing and Measuring Performance, will teach you how to analyze algorithmic complexity using big O notation. The chapter also discusses how to profile your code to find hotspots and how to set up performance tests using Google Benchmark.

Chapter 4, Data Structures, takes you through the importance of structuring data so that it can be accessed quickly. Containers from the standard library, such as `std::vector`, `std::list`, `std::unordered_map`, and `std::priority_queue`, are introduced. Finally, this chapter demonstrates how to use parallel arrays.

Chapter 5, Algorithms, introduces the most important algorithms from the standard library. You will also learn how to use iterators and ranges, and how to implement your own generic algorithms.

Chapter 6, Ranges and Views, will teach you how to compose algorithms using the Ranges library introduced in C++20. You will learn why views from the Ranges library are useful and some benefits of lazy evaluation.

Chapter 7, Memory Management, focuses on safe and efficient memory management. This includes memory ownership, RAII, smart pointers, stack memory, dynamic memory, and custom memory allocators.

Chapter 8, Compile-Time Programming, explains metaprogramming techniques using `constexpr`, `constexpr`, and type traits. You will also learn how to use C++20 concepts and the new Concepts library. Finally, it provides practical examples of metaprogramming use cases, such as reflection.

Chapter 9, Essential Utilities, will guide you through the Utilities library and how to benefit from types such as `std::optional`, `std::any`, and `std::variant` using compile-time programming techniques.

Chapter 10, Proxy Objects and Lazy Evaluation, explores how proxy objects can be used to perform under-the-hood optimizations while preserving clean syntax. Additionally, some creative uses of operator-overloading are demonstrated.

Chapter 11, Concurrency, covers the fundamentals of concurrent programming, including parallel execution, shared memory, data races, and deadlocks. It also includes an introduction to the C++ Thread support library, the Atomic library, and the C++ memory model.

Chapter 12, Coroutines and Lazy Generators, contains a general introduction to the coroutine abstraction. You will learn how ordinary functions and coroutines are executed on the CPU using the stack and the heap. C++20 stackless coroutines are introduced and you will discover how to solve problems using generators.

Chapter 13, Asynchronous Programming with Coroutines, introduces concurrent programming using stackless coroutines from C++20 and touches on the subject of asynchronous network programming using Boost.Asio.

Chapter 14, Parallel Algorithms, starts by showing the complexity of writing parallel algorithms and how to measure their performance. It then demonstrates how to utilize standard library algorithms in a parallel context using execution policies.

Get the most out of this book

To get the most out of this book, you need to have a basic knowledge of C++. It's preferable that you have already faced problems related to performance and are now looking for new tools and practices to have ready for the next time you need to work with performance and C++.

There are a lot of code examples in this book. Some are taken from the real world, but most of them are artificial or vastly simplified examples to prove a concept, rather than provide you with production-ready code.

I have put all the code examples in source files divided by chapter so that it is fairly easy to find the examples you want to experiment with. If you open up the source code files, you will note that I have replaced most of the `main()` functions from the examples with test cases written with the Google Test framework. I hope that this will help you rather than confuse you. It allowed me to write helpful descriptions for each example, and it also makes it easier to run all the examples from one chapter at once.

In order to compile and run the examples, you will need the following:

- A computer
- An operating system (the examples have been verified on Windows, Linux, and macOS)
- A compiler (I used Clang, GCC, and Microsoft Visual C++)
- CMake

The CMake script provided with the example code will download and install further dependencies, such as Boost, Google Benchmark, and Google Test.

During the writing of this book, I found it helpful to use **Compiler Explorer**, which is available at <https://godbolt.org/>. Compiler Explorer is an online compiler service that lets you try various compilers and versions. Try it out if you haven't already!

Download the example code files

The code bundle for the book is hosted on GitHub at <https://github.com/PacktPublishing/Cpp-High-Performance-Second-Edition>. If there's an update to the code, it will be updated on the existing GitHub repository.

There are other code bundles from Packt's rich catalog of books and videos available at <https://github.com/PacktPublishing/>. Check them out!

Download the color images

Packt also provides a PDF file that has color images of the screenshots/diagrams used in this book. You can download it here: https://static.packt-cdn.com/downloads/9781839216541_ColorImages.pdf.

Conventions used

There are a number of text conventions used throughout this book.

CodeInText : Indicates code words in text, folder names, filenames, file extensions, dummy URLs, and user input. Here is an example: "The keyword `constexpr` was introduced in C++11."

A block of code is set as follows:

```
#include <iostream>
int main() {
    std::cout << "High Performance C++\n";
}
```


When I wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:

```
#include <iostream>
int main() {
    std :: cout << "High Performance C++\n" ;
}
```

Any command-line input or output is written as follows:

```
$ clang++ -std=c++20 high_performance.cpp
$ ./a.out
$ High Performance C++
```

Bold: Indicates a new term, an important word, or words that you see on the screen. For example: "Fill in the form and click on the **Save** button."



Warnings or important notes appear like this.

Get in touch

Feedback from readers is always welcome.

General feedback: If you have questions about any aspect of this book, mention the book title in the subject of your message and email Packt at customercare@packtpub.com.

Errata: Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you have found a mistake in this book, we would be grateful if you could report this to us. Please visit www.packtpub.com/support/errata, select your book, click on the **Errata Submission Form** link, and enter the details.

Piracy: If you come across any illegal copies of our works in any form on the Internet, we would be grateful if you could provide us with the location address or website name. Please contact us at copyright@packt.com with a link to the material.

If you are interested in becoming an author: If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, please visit authors.packtpub.com.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make a purchase decision, we at Packt can understand what you think about our product, and our author can see your feedback on their book. Thank you!

For more information about Packt, please visit packt.com.

Download a free PDF copy of this book

Thanks for purchasing this book!

Do you like to read on the go but are unable to carry your print books everywhere? Is your eBook purchase not compatible with the device of your choice?

Don't worry, now with every Packt book you get a DRM-free PDF version of that book at no cost.

Read anywhere, any place, on any device. Search, copy, and paste code from your favorite technical books directly into your application.

The perks don't stop there, you can get exclusive access to discounts, newsletters, and great free content in your inbox daily

Follow these simple steps to get the benefits:

1. Scan the QR code or visit the link below



<https://packt.link/free-ebook/9781839216541>

2. Submit your proof of purchase
3. That's it! We'll send your free PDF and other benefits to your email directly

