

---

# Modern C Programming With Test Driven Development

## Code Better Sleep Better

---

Hands-On Embedded Programming with C++17

Effective Modern C++

Software Architecture with C++

Expert C++ Programming

Design Patterns in Modern C++

Discovering Modern C++

The The Modern C++ Challenge

Crafting Test-Driven Software with Python

C Programming: Test Your Skills

Working Effectively with Legacy Code

C Programming: Test Your Skills: Test Your Skills

Modern C++ Design

Expert C Programming

Beginning C++ Programming

Modern C++ Programming Cookbook

Test Driven Development for Embedded C

Embedded Programming with Modern C++ Cookbook

Modern C++ Programming Cookbook

Pragmatic Unit Testing in Java 8 with JUnit

Functional Programming in C++

Test-Driven Development with C++

Modern C++ Programming with Test-Driven Development

Beginning C++17

C++ Crash Course  
Mastering C++ Programming  
Effective C  
Modern Multithreading  
C Programming  
Modern C++ Programming Cookbook - Second Edition  
Cross-Platform Development with Qt 6 and Modern C++  
Modern C++ Programming Cookbook  
Introduction to Programming with C++ for Engineers  
Improving your C# Skills  
Murach's C++ Programming  
C Programming  
The C++ Programming Language  
Hands-On Functional Programming with C++  
Googletest in Practice  
C++ Crash Course  
Modern C++ Programming with Test-Driven Development

*Modern C Programming With Test  
Driven Development Code Better Sleep  
Better*

Downloaded from [aofithealth.com](http://aofithealth.com) by  
guest

---

## **SANAA KOCH**

---

**Hands-On Embedded Programming with C++17** Prentice  
Hall Professional

If you program in C++ you've been neglected. Test-driven development (TDD) is a modern software development practice that can dramatically reduce the number of defects in systems, produce more maintainable code, and give you the confidence to

change your software to meet changing needs. But C++ programmers have been ignored by those promoting TDD--until now. In this book, Jeff Langr gives you hands-on lessons in the challenges and rewards of doing TDD in C++. Modern C++ Programming With Test-Driven Development, the only comprehensive treatment on TDD in C++ provides you with everything you need to know about TDD, and the challenges and benefits of implementing it in your C++ systems. Its many detailed code examples take you step-by-step from TDD basics to advanced concepts. As a veteran C++ programmer, you're already writing high-quality code, and you work hard to maintain

code quality. It doesn't have to be that hard. In this book, you'll learn: how to use TDD to improve legacy C++ systems how to identify and deal with troublesome system dependencies how to do dependency injection, which is particularly tricky in C++ how to use testing tools for C++ that aid TDD new C++11 features that facilitate TDD As you grow in TDD mastery, you'll discover how to keep a massive C++ system from becoming a design mess over time, as well as particular C++ trouble spots to avoid. You'll find out how to prevent your tests from being a maintenance burden and how to think in TDD without giving up your hard-won C++ skills. Finally, you'll see how to grow and sustain TDD in your team. Whether you're a complete unit-testing novice or an experienced tester, this book will lead you to mastery of test-driven development in C++. What You Need A C++ compiler running under Windows or Linux, preferably one that supports C++11. Examples presented in the book were built under gcc 4.7.2. Google Mock 1.6 (downloadable for free; it contains Google Test as well) or an alternate C++ unit testing tool. Most examples in the book are written for Google Mock, but it isn't difficult to translate them to your tool of choice. A good programmer's editor or IDE. cmake, preferably. Of course, you can use your own preferred make too. CMakeLists.txt files are provided for each project. Examples provided were built using cmake version 2.8.9. Various freely-available third-party libraries are used as the basis for examples in the book. These include: - cURL- JsonCpp- Boost (filesystem, date\_time/gregorian, algorithm, assign)Several examples use the boost headers/libraries. Only one example uses cURL and JsonCpp.

[Effective Modern C++](#) Apress

Another day without Test-Driven Development means more time wasted chasing bugs and watching your code deteriorate. You thought TDD was for someone else, but it's not! It's for you, the embedded C programmer. TDD helps you prevent defects and build software with a long useful life. This is the first book to teach the hows and whys of TDD for C programmers. TDD is a modern programming practice C developers need to know. It's a different way to program---unit tests are written in a tight feedback loop with the production code, assuring your code does what you think. You get valuable feedback every few minutes. You find mistakes before they become bugs. You get early warning of design problems. You get immediate notification of side effect defects. You get to spend more time adding valuable features to your product. James is one of the few experts in applying TDD to embedded C. With his 1.5 decades of training, coaching, and practicing TDD in C, C++, Java, and C# he will lead you from being a novice in TDD to using the techniques that few have mastered. This book is full of code written for embedded C programmers. You don't just see the end product, you see code and tests evolve. James leads you through the thought process and decisions made each step of the way. You'll learn techniques for test-driving code right next to the hardware, and you'll learn design principles and how to apply them to C to keep your code clean and flexible. To run the examples in this book, you will need a C/C++ development environment on your machine, and the GNU GCC tool chain or Microsoft Visual Studio for C++ (some project conversion may be needed).

**Software Architecture with C++** Pragmatic Bookshelf

Test your C++ programming skills by solving real-world

programming problems covered in the book Key Features Solve a variety of real-world programming and logic problems by leveraging the power of C++17 Test your skills in using language features, algorithms, data structures, design patterns, and more Explore areas such as cryptography, communication, and image handling in C++ Book Description C++ is one of the most widely-used programming languages and has applications in a variety of fields, such as gaming, GUI programming, and operating systems, to name a few. Through the years, C++ has evolved into (and remains) one of the top choices for software developers worldwide. This book will show you some notable C++ features and how to implement them to meet your application needs. Each problem is unique and doesn't just test your knowledge of the language; it tests your ability to think out of the box and come up with the best solutions. With varying levels of difficulty, you'll be faced with a wide variety of challenges. And in case you're stumped, you don't have to worry: we've got the best solutions to the problems in the book. So are you up for the challenge? What you will learn Serialize and deserialize JSON and XML data Perform encryption and signing to facilitate secure communication between parties Embed and use SQLite databases in your applications Use threads and asynchronous functions to implement generic purpose parallel algorithms Compress and decompress files to/from a ZIP archive Implement data structures such as circular buffer and priority queue Implement general purpose algorithms as well as algorithms that solve specific problems Create client-server applications that communicate over TCP/IP Consume HTTP REST services Use design patterns to solve real-world problems Who this book is for

This book will appeal to C++ developers of all levels. There's a challenge inside for everyone.

Expert C++ Programming No Starch Press

A detailed introduction to the C programming language for experienced programmers. The world runs on code written in the C programming language, yet most schools begin the curriculum with Python or Java. Effective C bridges this gap and brings C into the modern era--covering the modern C17 Standard as well as potential C2x features. With the aid of this instant classic, you'll soon be writing professional, portable, and secure C programs to power robust systems and solve real-world problems. Robert C. Seacord introduces C and the C Standard Library while addressing best practices, common errors, and open debates in the C community. Developed together with other C Standards committee experts, Effective C will teach you how to debug, test, and analyze C programs. You'll benefit from Seacord's concise explanations of C language constructs and behaviors, and from his 40 years of coding experience. You'll learn: How to identify and handle undefined behavior in a C program The range and representations of integers and floating-point values How dynamic memory allocation works and how to use nonstandard functions How to use character encodings and types How to perform I/O with terminals and filesystems using C Standard streams and POSIX file descriptors How to understand the C compiler's translation phases and the role of the preprocessor How to test, debug, and analyze C programs Effective C will teach you how to write professional, secure, and portable C code that will stand the test of time and help strengthen the foundation of the computing world.

**Design Patterns in Modern C++** Pearson Education India  
Enhance your cross-platform programming abilities with the powerful features and capabilities of Qt 6 Key Features Leverage Qt and C++ capabilities to create modern, cross-platform applications that can run on a wide variety of software applications Explore what's new in Qt 6 and understand core concepts in depth Build professional customized GUI applications with the help of Qt Creator Book Description Qt is a cross-platform application development framework widely used for developing applications that can run on a wide range of hardware platforms with little to no change in the underlying codebase. If you have basic knowledge of C++ and want to build desktop or mobile applications with a modern graphical user interface (GUI), Qt is the right choice for you. Cross-Platform Development with Qt 6 and Modern C++ helps you understand why Qt is one of the favorite GUI frameworks adopted by industries worldwide, covering the essentials of programming GUI apps across a multitude of platforms using the standard C++17 and Qt 6 features. Starting with the fundamentals of the Qt framework, including the features offered by Qt Creator, this practical guide will show you how to create classic user interfaces using Qt Widgets and touch-friendly user interfaces using Qt Quick. As you advance, you'll explore the Qt Creator IDE for developing applications for multiple desktops as well as for embedded and mobile platforms. You will also learn advanced concepts about signals and slots. Finally, the book takes you through debugging and testing your app with Qt Creator IDE. By the end of this book, you'll be able to build cross-platform applications with a modern GUI along with the speed and power of native apps. What you will

learn Write cross-platform code using the Qt framework to create interactive applications Build a desktop application using Qt Widgets Create a touch-friendly user interface with Qt Quick Develop a mobile application using Qt and deploy it on different platforms Get to grips with Model/View programming with Qt Widgets and Qt Quick Discover Qt's graphics framework and add animations to your user interface Write test cases using the Qt Test framework and debug code Build a translation-aware application Follow best practices in Qt to write high-performance code Who this book is for This book is for application developers who want to use C++ and Qt to create modern, responsive applications that can be deployed to multiple operating systems such as Microsoft Windows, Apple macOS, and Linux desktop platforms. Although no prior knowledge of Qt is expected, beginner-level knowledge of the C++ programming language and object-oriented programming system (OOPs) concepts will be helpful.

*Discovering Modern C++* Addison-Wesley Professional

The Pragmatic Programmers classic is back! Freshly updated for modern software development, *Pragmatic Unit Testing in Java 8 With JUnit* teaches you how to write and run easily maintained unit tests in JUnit with confidence. You'll learn mnemonics to help you know what tests to write, how to remember all the boundary conditions, and what the qualities of a good test are. You'll see how unit tests can pay off by allowing you to keep your system code clean, and you'll learn how to handle the stuff that seems too tough to test. *Pragmatic Unit Testing in Java 8 With JUnit* steps you through all the important unit testing topics. If you've never written a unit test, you'll see screen shots from Eclipse,

IntelliJ IDEA, and NetBeans that will help you get past the hard part--getting set up and started. Once past the basics, you'll learn why you want to write unit tests and how to effectively use JUnit. But the meaty part of the book is its collected unit testing wisdom from people who've been there, done that on production systems for at least 15 years: veteran author and developer Jeff Langr, building on the wisdom of Pragmatic Programmers Andy Hunt and Dave Thomas. You'll learn: How to craft your unit tests to minimize your effort in maintaining them. How to use unit tests to help keep your system clean. How to test the tough stuff. Memorable mnemonics to help you remember what's important when writing unit tests. How to help your team reap and sustain the benefits of unit testing. You won't just learn about unit testing in theory--you'll work through numerous code examples. When it comes to programming, hands-on is the only way to learn!

**The Modern C++ Challenge** Pearson Deutschland GmbH  
Software -- Programming Languages.

*Crafting Test-Driven Software with Python* Packt Publishing Ltd  
Build safety-critical and memory-safe stand-alone and networked embedded systems  
Key Features  
Know how C++ works and compares to other languages used for embedded development  
Create advanced GUIs for embedded devices to design an attractive and functional UI  
Integrate proven strategies into your design for optimum hardware performance  
Book Description  
C++ is a great choice for embedded development, most notably, because it does not add any bloat, extends maintainability, and offers many advantages over different programming languages. Hands-On Embedded Programming with C++17 will show you how C++ can be used to build robust and

concurrent systems that leverage the available hardware resources. Starting with a primer on embedded programming and the latest features of C++17, the book takes you through various facets of good programming. You'll learn how to use the concurrency, memory management, and functional programming features of C++ to build embedded systems. You will understand how to integrate your systems with external peripherals and efficient ways of working with drivers. This book will also guide you in testing and optimizing code for better performance and implementing useful design patterns. As an additional benefit, you will see how to work with Qt, the popular GUI library used for building embedded systems. By the end of the book, you will have gained the confidence to use C++ for embedded programming. What you will learn  
Choose the correct type of embedded platform to use for a project  
Develop drivers for OS-based embedded systems  
Use concurrency and memory management with various microcontroller units (MCUs)  
Debug and test cross-platform code with Linux  
Implement an infotainment system using a Linux-based single board computer  
Extend an existing embedded system with a Qt-based GUI  
Communicate with the FPGA side of a hybrid FPGA/SoC system  
Who this book is for  
If you want to start developing effective embedded programs in C++, then this book is for you. Good knowledge of C++ language constructs is required to understand the topics covered in the book. No knowledge of embedded systems is assumed.

C Programming: Test Your Skills John Wiley & Sons

The most widely read and trusted guide to the C++ language, standard library, and design techniques includes significant new

updates and two new appendices on internationalization and Standard Library technicalities. It is the only book with authoritative, accessible coverage of every major element of ISO/ANSI Standard C++.

**Working Effectively with Legacy Code** Addison-Wesley Professional

Get to grips with essential concepts and step-by-step explanations to apply TDD practices to your Python projects while keeping your test suite under control

**Key Features**

- Build robust Python applications using TDD and BDD methodologies
- Test Python web applications using WebTest and web frameworks
- Leverage pytest to implement stringent testing mechanisms to ensure fault-tolerant applications

**Book Description**

Test-driven development (TDD) is a set of best practices that helps developers to build more scalable software and is used to increase the robustness of software by using automatic tests. This book shows you how to apply TDD practices effectively in Python projects. You'll begin by learning about built-in unit tests and Mocks before covering rich frameworks like pytest and web-based libraries such as WebTest and Robot Framework, discovering how Python allows you to embrace all modern testing practices with ease. Moving on, you'll find out how to design tests and balance them with new feature development and learn how to create a complete test suite with pytest. The book helps you adopt a hands-on approach to implementing TDD and associated methodologies that will have you up and running and make you more productive in no time. With the help of step-by-step explanations of essential concepts and practical examples, you'll explore automatic tests and TDD best practices and get to grips

with the methodologies and tools available in Python for creating effective and robust applications. By the end of this Python book, you will be able to write reliable test suites in Python to ensure the long-term resilience of your application using the range of libraries offered by Python for testing and development. What you will learn

Find out how tests can make your life easier as a developer and discover related best practices

Explore pytest, the most widespread testing framework for Python

Get to grips with the most common pytest plugins, including coverage, flaky, xdist, and picked

Write functional tests for WSGI web applications with WebTest

Run end-to-end tests for web applications using Robot Framework

Understand what test-driven development means and why it is important

Discover how to use the range of tools available in Python

Build reliable and robust applications

Who this book is for

This book is for Python developers looking to get started with test-driven development and developers who want to learn about the testing tools available in Python. Developers who want to create web applications with Python and plan to implement TDD methodology with pytest will find this book useful. Basic knowledge of Python programming is required.

*C Programming: Test Your Skills: Test Your Skills* Pragmatic Bookshelf

In the beginning, C++ was a hard language to learn because it required programmers to master low-level techniques to work with memory. Over the years, C++ has evolved to provide higher-level techniques that make it much easier to write effective code. But most C++ books haven't evolved with the language. Until now. Now, this book uses modern C++ to get you off to a fast start, and then builds out your coding and OOP skills

to the professional level. At that point, it also covers older techniques so you'll be able to maintain the vast amount of legacy code that's out there, as well as work with embedded systems that don't support the newer techniques.

*Modern C++ Design* Packt Publishing Ltd

C++ was written to help professional C# developers learn modern C++ programming. The aim of this book is to leverage your existing C# knowledge in order to expand your skills. Whether you need to use C++ in an upcoming project, or simply want to learn a new language (or reacquaint yourself with it), this book will help you learn all of the fundamental pieces of C++ so you can begin writing your own C++ programs. This updated and expanded second edition of the book provides a user-friendly introduction to the subject. Taking a clear structural framework, it guides the reader through the subject's core elements. A flowing writing style combines with the use of illustrations and diagrams throughout the text to ensure the reader understands even the most complex of concepts. This succinct and enlightening overview is a required reading for all those interested in the subject. We hope you find this book useful in shaping your future career & Business.

### **Expert C Programming** Pragmatic Bookshelf

As scientific and engineering projects grow larger and more complex, it is increasingly likely that those projects will be written in C++. With embedded hardware growing more powerful, much of its software is moving to C++, too. Mastering C++ gives you strong skills for programming at nearly every level, from "close to the hardware" to the highest-level abstractions. In short, C++ is a language that scientific and technical practitioners need to

know. Peter Gottschling's *Discovering Modern C++* is an intensive introduction that guides you smoothly to sophisticated approaches based on advanced features. Gottschling introduces key concepts using examples from many technical problem domains, drawing on his extensive experience training professionals and teaching C++ to students of physics, math, and engineering. This book is designed to help you get started rapidly and then master increasingly robust features, from lambdas to expression templates. You'll also learn how to take advantage of the powerful libraries available to C++ programmers: both the Standard Template Library (STL) and scientific libraries for arithmetic, linear algebra, differential equations, and graphs. Throughout, Gottschling demonstrates how to write clear and expressive software using object orientation, generics, metaprogramming, and procedural techniques. By the time you're finished, you'll have mastered all the abstractions you need to write C++ programs with exceptional quality and performance.

### *Beginning C++ Programming* Apress

Learn how to write a simple testing framework and extend it to drive the design of your logging library  
**Key Features:** Learn how to solve various challenges when testing in C++ with the help of effective solutions  
 Develop a logging library with enhancements  
 Drive better code designs with effective tests  
**Book Description:** Modern, standard C++ is all that is needed to create a small and practical testing framework that will improve the design of any project. This allows you to think about how the code will be used, which is the first step in designing intuitive interfaces. TDD is a modern balanced software development approach that helps to



create maintainable applications, provide modularity in design, and write minimal code that drastically reduces defects. With the help of this book, you'll be able to continue adding value when designs need to change by ensuring that the changes don't break existing tests. In this book, developers working with test-driven development (TDD) will be able to put their knowledge to work by writing a simple testing framework and then using it to drive the design of a logging library. The book will help you enhance your software development skills with test cases. You'll understand how to design and implement test cases. The chapters will also show you how to utilize the TDD approach to be more productive in software development than attempting to code in large unstructured steps. By the end of this book, you'll have gained knowledge of TDD and testing and also built a working logging library. What You Will Learn: Understand how to develop software using TDD Keep the code for the system as error-free as possible Refactor and redesign code confidently Communicate the requirements and behaviors of the code with your team Understand the differences between unit tests and integration tests Use TDD to create a minimal viable testing framework Who this book is for: This book is for C++ developers already familiar with and using C++ for daily tasks who want to improve their skillset. You don't need to be an expert but you should already have some knowledge of modern C++ and how to use templates to get the most out of this book.

*Modern C++ Programming Cookbook* Packt Publishing Ltd  
Learn functional programming and build robust applications using the latest functional features in C++ Key Features Learn programming concepts such as loops, expressive code, and

simple parallelization Understand the working of Lambdas and Currying and write Pure functions Explore event sourcing and other functional patterns to improve the efficiency of your applications Book Description Functional programming enables you to divide your software into smaller, reusable components that are easy to write, debug, and maintain. Combined with the power of C++, you can develop scalable and functional applications for modern software requirements. This book will help you discover the functional features in C++ 17 and C++ 20 to build enterprise-level applications. Starting with the fundamental building blocks of functional programming and how to use them in C++, you'll explore functions, currying, and lambdas. As you advance, you'll learn how to improve cohesion and delve into test-driven development, which will enable you in designing better software. In addition to this, the book covers architectural patterns such as event sourcing to help you get to grips with the importance of immutability for data storage. You'll even understand how to "think in functions" and implement design patterns in a functional way. By the end of this book, you'll be able to write faster and cleaner production code in C++ with the help of functional programming. What you will learn Understand the fundamentals of functional programming Structure your code by understanding the building blocks of functional programming Compare design styles in functional programming and object-oriented programming (OOP) Use the concept of currying to create new functions in C++ Become skilled at implementing design patterns in a functional way Get to grips with multithreading by means of functional programming Learn how to improve memory

consumption when using functional constructs Who this book is for This book is for C++ developers who want to learn functional programming but have little to no knowledge of the paradigm. Although no prior knowledge of functional programming is necessary, basic C++ programming experience will help you understand key concepts covered in the book.

Test Driven Development for Embedded C Packt Publishing Ltd  
Get more out of your legacy systems: more performance, functionality, reliability, and manageability Is your code easy to change? Can you get nearly instantaneous feedback when you do change it? Do you understand it? If the answer to any of these questions is no, you have legacy code, and it is draining time and money away from your development efforts. In this book, Michael Feathers offers start-to-finish strategies for working more effectively with large, untested legacy code bases. This book draws on material Michael created for his renowned Object Mentor seminars: techniques Michael has used in mentoring to help hundreds of developers, technical managers, and testers bring their legacy systems under control. The topics covered include Understanding the mechanics of software change: adding features, fixing bugs, improving design, optimizing performance Getting legacy code into a test harness Writing tests that protect you against introducing new problems Techniques that can be used with any language or platform—with examples in Java, C++, C, and C# Accurately identifying where code changes need to be made Coping with legacy systems that aren't object-oriented Handling applications that don't seem to have any structure This book also includes a catalog of twenty-four dependency-breaking techniques that help you work with program elements in isolation

and make safer changes.

*Embedded Programming with Modern C++ Cookbook* No Starch Press

Coming to grips with C++11 and C++14 is more than a matter of familiarizing yourself with the features they introduce (e.g., auto type declarations, move semantics, lambda expressions, and concurrency support). The challenge is learning to use those features effectively—so that your software is correct, efficient, maintainable, and portable. That's where this practical book comes in. It describes how to write truly great software using C++11 and C++14—i.e. using modern C++. Topics include: The pros and cons of braced initialization, noexcept specifications, perfect forwarding, and smart pointer make functions The relationships among `std::move`, `std::forward`, rvalue references, and universal references Techniques for writing clear, correct, effective lambda expressions How `std::atomic` differs from `volatile`, how each should be used, and how they relate to C++'s concurrency API How best practices in "old" C++ programming (i.e., C++98) require revision for software development in modern C++ Effective Modern C++ follows the proven guideline-based, example-driven format of Scott Meyers' earlier books, but covers entirely new material. "After I learned the C++ basics, I then learned how to use C++ in production code from Meyer's series of Effective C++ books. Effective Modern C++ is the most important how-to book for advice on key guidelines, styles, and idioms to use modern C++ effectively and well. Don't own it yet? Buy this one. Now". -- Herb Sutter, Chair of ISO C++ Standards Committee and C++ Software Architect at Microsoft

**Modern C++ Programming Cookbook** Packt Publishing Ltd

If you program in C++ you've been neglected. Test-driven development (TDD) is a modern software development practice that can dramatically reduce the number of defects in systems, produce more maintainable code, and give you the confidence to change your software to meet changing needs. But C++ programmers have been ignored by those promoting TDD--until now. In this book, Jeff Langr gives you hands-on lessons in the challenges and rewards of doing TDD in C++. Modern C++ Programming With Test-Driven Development, the only comprehensive treatment on TDD in C++ provides you with everything you need to know about TDD, and the challenges and benefits of implementing it in your C++ systems. Its many detailed code examples take you step-by-step from TDD basics to advanced concepts. As a veteran C++ programmer, you're already writing high-quality code, and you work hard to maintain code quality. It doesn't have to be that hard. In this book, you'll learn: how to use TDD to improve legacy C++ systems how to identify and deal with troublesome system dependencies how to do dependency injection, which is particularly tricky in C++ how to use testing tools for C++ that aid TDD new C++11 features that facilitate TDD As you grow in TDD mastery, you'll discover how to keep a massive C++ system from becoming a design mess over time, as well as particular C++ trouble spots to avoid. You'll find out how to prevent your tests from being a maintenance burden and how to think in TDD without giving up your hard-won C++ skills. Finally, you'll see how to grow and sustain TDD in your team. Whether you're a complete unit-testing novice or an experienced tester, this book will lead you to mastery of test-driven development in C++. What You Need A

C++ compiler running under Windows or Linux, preferably one that supports C++11. Examples presented in the book were built under gcc 4.7.2. Google Mock 1.6 (downloadable for free; it contains Google Test as well) or an alternate C++ unit testing tool. Most examples in the book are written for Google Mock, but it isn't difficult to translate them to your tool of choice. A good programmer's editor or IDE. cmake, preferably. Of course, you can use your own preferred make too. CMakeLists.txt files are provided for each project. Examples provided were built using cmake version 2.8.9. Various freely-available third-party libraries are used as the basis for examples in the book. These include: cURL JsonCpp Boost (filesystem, date\_time/gregorian, algorithm, assign) Several examples use the boost headers/libraries. Only one example uses cURL and JsonCpp.

Pragmatic Unit Testing in Java 8 with JUnit Packt Publishing Ltd Apply business requirements to IT infrastructure and deliver a high-quality product by understanding architectures such as microservices, DevOps, and cloud-native using modern C++ standards and features Key FeaturesDesign scalable large-scale applications with the C++ programming languageArchitect software solutions in a cloud-based environment with continuous integration and continuous delivery (CI/CD)Achieve architectural goals by leveraging design patterns, language features, and useful toolsBook Description Software architecture refers to the high-level design of complex applications. It is evolving just like the languages we use, but there are architectural concepts and patterns that you can learn to write high-performance apps in a high-level language without sacrificing readability and maintainability. If you're working with modern C++, this practical

guide will help you put your knowledge to work and design distributed, large-scale apps. You'll start by getting up to speed with architectural concepts, including established patterns and rising trends, then move on to understanding what software architecture actually is and start exploring its components. Next, you'll discover the design concepts involved in application architecture and the patterns in software development, before going on to learn how to build, package, integrate, and deploy your components. In the concluding chapters, you'll explore different architectural qualities, such as maintainability, reusability, testability, performance, scalability, and security. Finally, you will get an overview of distributed systems, such as service-oriented architecture, microservices, and cloud-native, and understand how to apply them in application development. By the end of this book, you'll be able to build distributed services using modern C++ and associated tools to deliver solutions as per your clients' requirements. What you will learn Understand

how to apply the principles of software architecture Apply design patterns and best practices to meet your architectural goals Write elegant, safe, and performant code using the latest C++ features Build applications that are easy to maintain and deploy Explore the different architectural approaches and learn to apply them as per your requirement Simplify development and operations using application containers Discover various techniques to solve common problems in software design and development Who this book is for This software architecture C++ programming book is for experienced C++ developers looking to become software architects or develop enterprise-grade applications.

**Functional Programming in C++** Simon and Schuster  
C Programming: Test Your Skills is specifically designed to be used as the supplementary resource for learning C Programming. It is ideal for self practice or test preparation and hones one's problem solving abilities through varieties of exercises

Best Sellers - Books :

- [Taylor Swift: A Little Golden Book Biography By Wendy Loggia](#)
- [Stop Overthinking: 23 Techniques To Relieve Stress, Stop Negative Spirals, Declutter Your Mind, And Focus On The Present \(the Path To Calm\) By Nick Trenton](#)
- [Remarkably Bright Creatures: A Read With Jenna Pick](#)
- [Regretting You By Colleen Hoover](#)
- [Tucker By Chadwick Moore](#)
- [If He Had Been With Me](#)
- [Playground](#)
- [The Mountain Is You: Transforming Self-sabotage Into Self-mastery](#)
- [Young Forever: The Secrets To Living Your Longest, Healthiest Life \(the Dr. Hyman Library, 11\) By Dr. Mark Hyman Md](#)

- [Little Blue Truck's Springtime: An Easter And Springtime Book For Kids By Alice Schertle](#)