

C Concurrency In Action Practical Multithreading

Master multithreading and concurrent processing with C++ About This Book Delve into the fundamentals of multithreading and concurrency and find out how to implement them Explore atomic operations to optimize code performance Apply concurrency to both distributed computing and GPGPU processing Who This Book Is For This book is for intermediate C++ developers who wish to extend their knowledge of multithreading and concurrent processing. You should have basic experience with multithreading and be comfortable using C++ development toolchains on the command line. What You Will Learn Deep dive into the details of the how various operating systems currently implement multithreading Choose the best multithreading APIs when designing a new application Explore the use of mutexes, spin-locks, and other synchronization concepts and see how to safely pass data between threads Understand the level of API support provided by various C++ toolchains Resolve common issues in multithreaded code and recognize common pitfalls using tools such as Memcheck, CacheGrind, DRD, Helgrind, and more Discover the nature of atomic operations and understand how they can be useful in optimizing code Implement a multithreaded application in a distributed computing environment Design a C++-based GPGPU application that employs multithreading In Detail Multithreaded applications execute multiple threads in a single processor environment, allowing developers achieve concurrency. This book will teach you the finer points of multithreading and concurrency concepts and how to apply them efficiently in C++. Divided into three modules, we start with a brief introduction to the fundamentals of multithreading and concurrency concepts. We then take an in-depth look at how these concepts work at the hardware-level as well as how both operating systems and frameworks use these low-level functions. In the next module, you will learn about the native multithreading and concurrency support available in C++ since the 2011 revision, synchronization and communication between threads, debugging concurrent C++ applications, and the best programming practices in C++. In the final module, you will learn about atomic operations before moving on to apply concurrency to distributed and GPGPU-based processing. The comprehensive coverage of essential multithreading concepts means you will be able to efficiently apply multithreading concepts while coding in C++. Style and approach This book is filled with examples that will help you become a master at writing robust concurrent and parallel applications in C++.

The Art of Multiprocessor Programming, Second Edition, provides users with an authoritative guide to multicore programming. This updated edition introduces higher level software development skills relative to those needed for efficient single-core programming, and includes comprehensive coverage of the new principles, algorithms, and tools necessary for effective multiprocessor programming. The book is an ideal resource for students and professionals alike who will benefit from its thorough coverage of key multiprocessor programming issues. Features new exercises developed for instructors using the text, with more algorithms, new examples, and other updates throughout the book Presents the fundamentals of programming multiple threads for accessing shared memory Explores mainstream concurrent data structures and the key elements of their design, as well as synchronization techniques, from simple locks to transactional memory systems

Summary Revised and updated for Elixir 1.7, Elixir in Action, Second Edition teaches you how to apply Elixir to practical problems associated with scalability, fault tolerance, and high availability. Along the way, you'll develop an appreciation for, and considerable skill in, a functional and concurrent style of programming. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology When you're building mission-critical software, fault tolerance matters. The Elixir programming language delivers fast, reliable applications, whether you're building a large-scale distributed system, a set of backend services, or a simple web app. And Elixir's elegant syntax and functional programming mindset make your software easy to write, read, and maintain. About the Book Elixir in Action, Second Edition teaches you how to build production-quality distributed applications using the Elixir programming language. Author Saša Jurišić introduces this powerful language using examples that highlight the benefits of Elixir's functional and concurrent programming. You'll discover how the OTP framework can radically reduce tedious low-level coding tasks. You'll also explore practical approaches to concurrency as you learn to distribute a production system over multiple machines. What's inside Updated for Elixir 1.7 Functional and concurrent programming Introduction to distributed system design Creating deployable releases About the Reader You'll need intermediate skills with client/server applications and a language like Java, C#, or Ruby. No previous experience with Elixir required. About the Author Saša Jurišić is a developer with extensive experience using Elixir and Erlang in complex server-side systems. Table of Contents First steps Building blocks Control flow Data abstractions Concurrency primitives Generic server processes Building a concurrent system Fault-tolerance basics Isolating error effects Beyond GenServer Working with components Building a distributed system Running the system

'Downright revolutionary... the title is a major understatement... 'Quantum Programming' may ultimately change the way embedded software is designed.' -- Michael Barr, Editor-in-Chief, Embedded Systems Programming magazine (Click here

If you're one of the many developers uncertain about concurrent and multithreaded development, this practical cookbook will change your mind. With more than 75 code-rich recipes, author Stephen Cleary demonstrates parallel processing and asynchronous programming techniques, using libraries and language features in .NET 4.5 and C# 5.0. Concurrency is becoming more common in responsive and scalable application development, but it's been extremely difficult to code. The detailed solutions in this cookbook show you how modern tools raise the level of abstraction, making concurrency much easier than before.

Complete with ready-to-use code and discussions about how and why the solution works, you get recipes for using: async and await for asynchronous operations Parallel programming with the Task Parallel Library The TPL Dataflow library for creating dataflow pipelines Capabilities that Reactive Extensions build on top of LINQ Unit testing with concurrent code Interop scenarios for combining concurrent approaches Immutable, threadsafe, and producer/consumer collections Cancellation support in your concurrent code Asynchronous-friendly Object-Oriented Programming Thread synchronization for accessing data

The C++11 standard allows programmers to express ideas more clearly, simply, and directly, and to write faster, more efficient code. Bjarne Stroustrup, the designer and original implementer of C++, thoroughly covers the details of this language and its use in his definitive reference, The C++ Programming Language, Fourth Edition. In A Tour of C++ , Stroustrup excerpts the overview chapters from that complete reference, expanding and enhancing them to give an experienced programmer—in just a few hours—a clear idea of what constitutes modern C++. In this concise, self-contained guide, Stroustrup covers most major language features and the major standard-library components—not, of course, in great depth, but to a level that gives programmers a meaningful overview of the language, some key examples, and practical help in getting started. Stroustrup presents the C++ features in the context of the programming styles they support, such as object-oriented and generic programming. His tour is remarkably

comprehensive. Coverage begins with the basics, then ranges widely through more advanced topics, including many that are new in C++11, such as move semantics, uniform initialization, lambda expressions, improved containers, random numbers, and concurrency. The tour ends with a discussion of the design and evolution of C++ and the extensions added for C++11. This guide does not aim to teach you how to program (see Stroustrup's Programming: Principles and Practice Using C++ for that); nor will it be the only resource you'll need for C++ mastery (see Stroustrup's The C++ Programming Language, Fourth Edition, for that). If, however, you are a C or C++ programmer wanting greater familiarity with the current C++ language, or a programmer versed in another language wishing to gain an accurate picture of the nature and benefits of modern C++, you can't find a shorter or simpler introduction than this tour provides.

Push the limits of what C - and you - can do, with this high-intensity guide to the most advanced capabilities of C Key Features Make the most of C's low-level control, flexibility, and high performance A comprehensive guide to C's most powerful and challenging features A thought-provoking guide packed with hands-on exercises and examples Book Description There's a lot more to C than knowing the language syntax. The industry looks for developers with a rigorous, scientific understanding of the principles and practices. Extreme C will teach you to use C's advanced low-level power to write effective, efficient systems. This intensive, practical guide will help you become an expert C programmer. Building on your existing C knowledge, you will master preprocessor directives, macros, conditional compilation, pointers, and much more. You will gain new insight into algorithm design, functions, and structures. You will discover how C helps you squeeze maximum performance out of critical, resource-constrained applications. C still plays a critical role in 21st-century programming, remaining the core language for precision engineering, aviations, space research, and more. This book shows how C works with Unix, how to implement OO principles in C, and fully covers multi-processing. In Extreme C, Amini encourages you to think, question, apply, and experiment for yourself. The book is essential for anybody who wants to take their C to the next level. What you will learn Build advanced C knowledge on strong foundations, rooted in first principles Understand memory structures and compilation pipeline and how they work, and how to make most out of them Apply object-oriented design principles to your procedural C code Write low-level code that's close to the hardware and squeezes maximum performance out of a computer system Master concurrency, multithreading, multi-processing, and integration with other languages Unit Testing and debugging, build systems, and inter-process communication for C programming Who this book is for Extreme C is for C programmers who want to dig deep into the language and its capabilities. It will help you make the most of the low-level control C gives you.

Summary This bestseller has been updated and revised to cover all the latest changes to C++ 14 and 17! C++ Concurrency in Action, Second Edition teaches you everything you need to write robust and elegant multithreaded applications in C++17. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology You choose C++ when your applications need to run fast. Well-designed concurrency makes them go even faster. C++ 17 delivers strong support for the multithreaded, multiprocessor programming required for fast graphic processing, machine learning, and other performance-sensitive tasks. This exceptional book unpacks the features, patterns, and best practices of production-grade C++ concurrency. About the Book C++ Concurrency in Action, Second Edition is the definitive guide to writing elegant multithreaded applications in C++. Updated for C++ 17, it carefully addresses every aspect of concurrent development, from starting new threads to designing fully functional multithreaded algorithms and data structures. Concurrency master Anthony Williams presents examples and practical tasks in every chapter, including insights that will delight even the most experienced developer. What's inside Full coverage of new C++ 17 features Starting and managing threads Synchronizing concurrent operations Designing concurrent code Debugging multithreaded applications About the Reader Written for intermediate C and C++ developers. No prior experience with concurrency required. About the Author Anthony Williams has been an active member of the BSI C++ Panel since 2001 and is the developer of the just::thread Pro extensions to the C++ 11 thread library. Table of Contents Hello, world of concurrency in C++! Managing threads Sharing data between threads Synchronizing concurrent operations The C++ memory model and operations on atomic types Designing lock-based concurrent data structures Designing lock-free concurrent data structures Designing concurrent code Advanced thread management Parallel algorithms Testing and debugging multithreaded applications

If you're looking to take full advantage of multi-core processors with concurrent programming, this practical book provides the knowledge and hands-on experience you need. The Art of Concurrency is one of the few resources to focus on implementing algorithms in the shared-memory model of multi-core processors, rather than just theoretical models or distributed-memory architectures. The book provides detailed explanations and usable samples to help you transform algorithms from serial to parallel code, along with advice and analysis for avoiding mistakes that programmers typically make when first attempting these computations. Written by an Intel engineer with over two decades of parallel and concurrent programming experience, this book will help you: Understand parallelism and concurrency Explore differences between programming for shared-memory and distributed-memory Learn guidelines for designing multithreaded applications, including testing and tuning Discover how to make best use of different threading libraries, including Windows threads, POSIX threads, OpenMP, and Intel Threading Building Blocks Explore how to implement concurrent algorithms that involve sorting, searching, graphs, and other practical computations The Art of Concurrency shows you how to keep algorithms scalable to take advantage of new processors with even more cores. For developing parallel code algorithms for concurrent programming, this book is a must.

C++ High Performance, Second Edition enables you to measure and identify bottlenecks in the code and eradicate them to amplify your application's working speed without compromising the readability of your C++ codebase

With the new C++ Standard and Technical Report 2 (TR2), multi-threading is coming to C++ in a big way. TR2 will provide higher-level synchronization facilities that allow for a much greater level of abstraction, and make programming multi-threaded applications simpler and safer. Concurrent programming is required if programmers are to take advantage of the multi-core microprocessors increasingly available from Intel and others. The new standard for C++ has extensions to the language that make concurrent programming more accessible to regular developers. As a guide and reference to the new concurrency features in the upcoming C++ Standard and TR2, this book is invaluable for existing programmers familiar with writing multi-threaded code in C++ using platform-specific APIs, or in other languages, as well as C++ programmers who have never written multithreaded code before.

This book breaks down the C++ STL, teaching you how to extract its gems and apply them to your programming. About This Book Boost your productivity as a C++ developer with the latest features of C++17 Develop high-quality, fast, and portable applications with the varied features of the STL Migrate from older versions (C++11, C++14) to C++17 Who This Book Is For This book is for developers who would like to master the C++ STL and make full use of its components. Prior C++ knowledge is assumed. What You Will Learn Make your own iterator types, allocators, and thread pools. Master every standard container and every standard algorithm. Improve your code by replacing new/delete with smart pointers. Understand the difference between monomorphic algorithms, polymorphic algorithms, and generic algorithms. Learn the

meaning and applications of vocabulary type, product type and sum type. In Detail Modern C++ has come a long way since 2011. The latest update, C++17, has just been ratified and several implementations are on the way. This book is your guide to the C++ standard library, including the very latest C++17 features. The book starts by exploring the C++ Standard Template Library in depth. You will learn the key differences between classical polymorphism and generic programming, the foundation of the STL. You will also learn how to use the various algorithms and containers in the STL to suit your programming needs. The next module delves into the tools of modern C++. Here you will learn about algebraic types such as `std::optional`, vocabulary types such as `std::function`, smart pointers, and synchronization primitives such as `std::atomic` and `std::mutex`. In the final module, you will learn about C++'s support for regular expressions and file I/O. By the end of the book you will be proficient in using the C++17 standard library to implement real programs, and you'll have gained a solid understanding of the library's own internals. Style and approach This book takes a concise but comprehensive approach to explaining and applying the C++ STL, one feature at a time.

C++ Concurrency in Action, Second Edition is the definitive guide to writing elegant multithreaded applications in C++. Updated for C++ 17, it carefully addresses every aspect of concurrent development, from starting new threads to designing fully functional multithreaded algorithms and data structures. Concurrency master Anthony Williams presents examples and practical tasks in every chapter, including insights that will delight even the most experienced developer. -- Provided by publisher.

Software -- Programming Languages.

"This book should be on every C++ programmer's desk. It's clear, concise, and valuable." Rob Green, Bowling Green State University This bestseller has been updated and revised to cover all the latest changes to C++ 14 and 17! C++ Concurrency in Action, Second Edition teaches you everything you need to write robust and elegant multithreaded applications in C++17. You choose C++ when your applications need to run fast. Well-designed concurrency makes them go even faster. C++ 17 delivers strong support for the multithreaded, multiprocessor programming required for fast graphic processing, machine learning, and other performance-sensitive tasks. This exceptional book/course unpacks the features, patterns, and best practices of production-grade C++ concurrency. C++ Concurrency in Action, Second Edition is the definitive guide to writing elegant multithreaded applications in C++. Updated for C++ 17, it carefully addresses every aspect of concurrent development, from starting new threads to designing fully functional multithreaded algorithms and data structures. Concurrency master Anthony Williams presents examples and practical tasks in every chapter, including insights that will delight even the most experienced developer. Inside: Full coverage of new C++ 17 features Starting and managing threads Synchronizing concurrent operations Designing concurrent code Debugging multithreaded applications This book/course is written for intermediate C and C++ developers. No prior experience with concurrency required. Anthony Williams has been an active member of the BSI C++ Panel since 2001 and is the developer of the `just::thread` Pro extensions to the C++ 11 thread library. A thorough presentation of C++ concurrency capabilities. Maurizio Tomasi, University of Milan Highly recommended for programmers who want to further their knowledge of the latest C++ standard. Frédéric Flayol, 4Pro Web C++ The guide contains snippets for everyday use in your own projects and to help take your concurrency C++ skills from the Padawan to the Jedi level. Jura Shikin, IVI Technologies NARRATED BY LISA FARINA AND DIANA GARDINER.

Corpus linguistics is a research approach to investigate the patterns of language use empirically, based on analysis of large collections of natural texts. While corpus-based analysis has had relatively little influence on theoretical linguistics, it has revolutionized the study of language variation and use: what speakers and writers actually do with the lexical and grammatical resources of a language. Corpus-based research employs the research methods of quantitative and qualitative social science to investigate language use patterns empirically. This four-volume collection is organized around linguistic research questions that can be investigated from a corpus perspective and includes amongst others studies of individual words, comparisons of supposedly synonymous words, studies of grammatical variation, and sociolinguistic studies of dialects, registers, styles, and world varieties. Corpus-based analysis has also proven to be important for the study of historical change.

Provides information on building concurrent applications using Java.

Get to grips with modern software demands by learning the effective uses of Rust's powerful memory safety. Key Features Learn and improve the sequential performance characteristics of your software Understand the use of operating system processes in a high-scale concurrent system Learn of the various coordination methods available in the Standard library Book Description Most programming languages can really complicate things, especially with regard to unsafe memory access. The burden on you, the programmer, lies across two domains: understanding the modern machine and your language's pain-points. This book will teach you to how to manage program performance on modern machines and build fast, memory-safe, and concurrent software in Rust. It starts with the fundamentals of Rust and discusses machine architecture concepts. You will be taken through ways to measure and improve the performance of Rust code systematically and how to write collections with confidence. You will learn about the Sync and Send traits applied to threads, and coordinate thread execution with locks, atomic primitives, data-parallelism, and more. The book will show you how to efficiently embed Rust in C++ code and explore the functionalities of various crates for multithreaded applications. It explores implementations in depth. You will know how a mutex works and build several yourself. You will master radically different approaches that exist in the ecosystem for structuring and managing high-scale systems. By the end of the book, you will feel comfortable with designing safe, consistent, parallel, and high-performance applications in Rust. What you will learn Probe your programs for performance and accuracy issues Create your own threading and multi-processing environment in Rust Use coarse locks from Rust's Standard library Solve common synchronization problems or avoid synchronization using atomic programming Build lock-free/wait-free structures in Rust and understand their implementations in the crates ecosystem Leverage Rust's memory model and type system to build safety properties into your parallel programs Understand the new features of the Rust programming language to ease the writing of parallel programs Who this book is for This book is aimed at software engineers with a basic understanding of Rust who want to exploit the parallel and concurrent nature of modern computing environments, safely.

Summary Concurrency in .NET teaches you how to build concurrent and scalable programs in .NET using the functional paradigm. This intermediate-level guide is aimed at developers, architects, and passionate computer programmers who are interested in writing code with improved speed and effectiveness by adopting a declarative and pain-free programming style. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology Unlock the incredible performance built into your multi-processor machines. Concurrent applications run faster because they spread work across processor cores, performing several tasks at the same time. Modern tools and techniques on the .NET platform, including parallel LINQ, functional programming, asynchronous programming, and the Task Parallel Library, offer powerful alternatives to traditional thread-based concurrency. About the Book Concurrency in .NET teaches you to write code that delivers the speed you need for performance-sensitive applications. Featuring examples in both C# and F#, this book guides you through concurrent and parallel designs that emphasize functional programming in theory and practice. You'll start with the foundations of concurrency and master essential techniques and design practices to optimize code running on modern multiprocessor systems. What's Inside The

most important concurrency abstractions Employing the agent programming model Implementing real-time event-stream processing Executing unbounded asynchronous operations Best concurrent practices and patterns that apply to all platforms About the Reader For readers skilled with C# or F#. About the Book Riccardo Terrell is a seasoned software engineer and Microsoft MVP who is passionate about functional programming. He has over 20 years' experience delivering cost-effective technology solutions in a competitive business environment. Table of Contents PART 1 - Benefits of functional programming applicable to concurrent programs Functional concurrency foundations Functional programming techniques for concurrency Functional data structures and immutability PART 2 - How to approach the different parts of a concurrent program The basics of processing big data: data parallelism, part 1 PLINQ and MapReduce: data parallelism, part 2 Real-time event streams: functional reactive programming Task-based functional parallelism Task asynchronicity for the win Asynchronous functional programming in F# Functional combinators for fluent concurrent programming Applying reactive programming everywhere with agents Parallel workflow and agent programming with TPL Dataflow PART 3 - Modern patterns of concurrent programming applied Recipes and design patterns for successful concurrent programming Building a scalable mobile app with concurrent functional programming Provides a straightforward and practical approach to object-oriented concepts, analysis, design and programming for students on Higher National and degree courses.

Advanced Metaprogramming in Classic C++ aims to be both an introduction and a reference to C++ template metaprogramming (TMP); TMP is presented in the book as a set of techniques that will bring a new style in C++ and make code exceptionally clear and efficient. The book deals with language aspects, design patterns, examples and applications (seen as case studies). Special emphasis is put on small reusable techniques that will improve the quality of daily work. What makes the book exceptional is the level of understanding of the concepts involved imparted by the author. This is not just a rote overview of metaprogramming. You will truly understand difficult topics like static assertions, how to write metafunctions, overload resolution, lambda expressions, and many others. More than that, you will work through them with practical examples guided by the author's frank explanations. This book requires you to think and to learn and to understand the language so that you can program at a higher level.

The Definitive Guide to Windows API Programming, Fully Updated for Windows 7, Windows Server 2008, and Windows Vista Windows System Programming, Fourth Edition, now contains extensive new coverage of 64-bit programming, parallelism, multicore systems, and many other crucial topics. Johnson Hart's robust code examples have been updated and streamlined throughout. They have been debugged and tested in both 32-bit and 64-bit versions, on single and multiprocessor systems, and under Windows 7, Vista, Server 2008, and Windows XP. To clarify program operation, sample programs are now illustrated with dozens of screenshots. Hart systematically covers Windows externals at the API level, presenting practical coverage of all the services Windows programmers need, and emphasizing how Windows functions actually behave and interact in real-world applications. Hart begins with features used in single-process applications and gradually progresses to more sophisticated functions and multithreaded environments. Topics covered include file systems, memory management, exceptions, processes, threads, synchronization, interprocess communication, Windows services, and security. New coverage in this edition includes Leveraging parallelism and maximizing performance in multicore systems Promoting source code portability and application interoperability across Windows, Linux, and UNIX Using 64-bit address spaces and ensuring 64-bit/32-bit portability Improving performance and scalability using threads, thread pools, and completion ports Techniques to improve program reliability and performance in all systems Windows performance-enhancing API features available starting with Windows Vista, such as slim reader/writer locks and condition variables A companion Web site, jmhartsoftware.com, contains all sample code, Visual Studio projects, additional examples, errata, reader comments, and Windows commentary and discussion.

Over 90 recipes that leverage the powerful features of the Standard Library in C++17 About This Book Learn the latest features of C++ and how to write better code by using the Standard Library (STL). Reduce the development time for your applications. Understand the scope and power of STL features to deal with real-world problems. Compose your own algorithms without forfeiting the simplicity and elegance of the STL way. Who This Book Is For This book is for intermediate-to-advanced C++ programmers who want to get the most out of the Standard Template Library of the newest version of C++: C++ 17. What You Will Learn Learn about the new core language features and the problems they were intended to solve Understand the inner workings and requirements of iterators by implementing them Explore algorithms, functional programming style, and lambda expressions Leverage the rich, portable, fast, and well-tested set of well-designed algorithms provided in the STL Work with strings the STL way instead of handcrafting C-style code Understand standard support classes for concurrency and synchronization, and how to put them to work Use the filesystem library addition available with the C++17 STL In Detail C++ has come a long way and is in use in every area of the industry. Fast, efficient, and flexible, it is used to solve many problems. The upcoming version of C++ will see programmers change the way they code. If you want to grasp the practical usefulness of the C++17 STL in order to write smarter, fully portable code, then this book is for you. Beginning with new language features, this book will help you understand the language's mechanics and library features, and offers insight into how they work. Unlike other books, ours takes an implementation-specific, problem-solution approach that will help you quickly overcome hurdles. You will learn the core STL concepts, such as containers, algorithms, utility classes, lambda expressions, iterators, and more, while working on practical real-world recipes. These recipes will help you get the most from the STL and show you how to program in a better way. By the end of the book, you will be up to date with the latest C++17 features and save time and effort while solving tasks elegantly using the STL. Style and approach This recipe-based guide will show you how to make the best use of C++ together with the STL to squeeze more out of the standard language

Here is a programmer's guide to using and programming POSIX threads, commonly known as Pthreads. A "coder's book", this title tells how to use Pthreads in the real world, making efficient and portable applications. Pthreads are an important set of current tools programmers need to have in today's network-intensive climate.

Unlock the groundbreaking advances of deep learning with this extensively revised new edition of the bestselling original. Learn directly from the creator of Keras and master practical Python deep learning techniques that are easy to apply in the real world. In Deep Learning with Python, Second Edition you will learn: Deep learning from first principles Image classification and image segmentation Timeseries forecasting Text classification and machine translation Text generation, neural style transfer, and image generation Deep Learning with Python has taught thousands of readers how to put the full capabilities of deep learning into action. This extensively revised second edition introduces deep learning

using Python and Keras, and is loaded with insights for both novice and experienced ML practitioners. You'll learn practical techniques that are easy to apply in the real world, and important theory for perfecting neural networks. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the technology Recent innovations in deep learning unlock exciting new software capabilities like automated language translation, image recognition, and more. Deep learning is quickly becoming essential knowledge for every software developer, and modern tools like Keras and TensorFlow put it within your reach—even if you have no background in mathematics or data science. This book shows you how to get started. About the book Deep Learning with Python, Second Edition introduces the field of deep learning using Python and the powerful Keras library. In this revised and expanded new edition, Keras creator François Chollet offers insights for both novice and experienced machine learning practitioners. As you move through this book, you'll build your understanding through intuitive explanations, crisp illustrations, and clear examples. You'll quickly pick up the skills you need to start developing deep-learning applications. What's inside Deep learning from first principles Image classification and image segmentation Time series forecasting Text classification and machine translation Text generation, neural style transfer, and image generation About the reader For readers with intermediate Python skills. No previous experience with Keras, TensorFlow, or machine learning is required. About the author François Chollet is a software engineer at Google and creator of the Keras deep-learning library. Table of Contents 1 What is deep learning? 2 The mathematical building blocks of neural networks 3 Introduction to Keras and TensorFlow 4 Getting started with neural networks: Classification and regression 5 Fundamentals of machine learning 6 The universal workflow of machine learning 7 Working with Keras: A deep dive 8 Introduction to deep learning for computer vision 9 Advanced deep learning for computer vision 10 Deep learning for timeseries 11 Deep learning for text 12 Generative deep learning 13 Best practices for the real world 14 Conclusions

Since the introduction of Hoares' Communicating Sequential Processes notation, powerful new tools have transformed CSP into a practical way of describing industrial-sized problems. This book gives you the fundamental grasp of CSP concepts you'll need to take advantage of those tools. Part I provides a detailed foundation for working with CSP, using as little mathematics as possible. It introduces the ideas behind operational, denotational and algebraic models of CSP. Parts II and III go into greater detail about theory and practice. Topics include: parallel operators, hiding and renaming, piping and enslavement, buffers and communication, termination and sequencing, and semantic theory. Three detailed practical case studies are also presented. For anyone interested in modeling sequential processes.

Using the implementation of a deep learning framework as an example, C++ Template Metaprogramming in Practice: A Deep Learning Framework explains the application of metaprogramming in a relatively large project and emphasizes ways to optimize systems performance. The book is suitable for developers with a basic knowledge of C++. Developers familiar with mainstream deep learning frameworks can also refer to this book to compare the differences between the deep learning framework implemented with metaprogramming and compile-time computing with deep learning frameworks using object-oriented methods. Consisting of eight chapters, the book starts with two chapters discussing basic techniques of metaprogramming and compile-time computing. The rest of the book's chapters focus on the practical application of metaprogramming in a deep learning framework. It examines rich types and systems, expression templates, and writing complex meta-functions, as well as such topics as: Heterogeneous dictionaries and policy templates An introduction to deep learning Type system and basic data types Operations and expression templates Basic layers Composite and recurrent layers Evaluation and its optimization Metaprogramming can construct flexible and efficient code. For C++ developers who are familiar with object-oriented programming, the main difficulty in learning and mastering C++ metaprogramming is establishing the thinking mode of functional programming. The meta-programming approach involved at compile time is functional, which means that the intermediate results of the construction cannot be changed, and the impact may be greater than expected. This book enables C++ programmers to develop a functional mindset and metaprogramming skills. The book also discusses the development cost and use cost of metaprogramming and provides workarounds for minimizing these costs.

Over 60 recipes to help you create ultra-fast multithreaded applications using C++ with rules, guidelines, and best practices Overview Create multithreaded applications using the power of C++ Upgrade your applications with parallel execution in easy-to-understand steps Stay up to date with new Windows 8 concurrent tasks Avoid classical synchronization problems Understand Windows API and concurrent execution What you will learn from this book Use an object-oriented programming model with inheritance, overloading, and polymorphism Solve common Interprocess Communication problems and avoid deadlocks or starvation problems in your application development Manage threads efficiently using the CThread class Explore .NET CLI/C++ features as well as synchronization objects and techniques Make use of parallel techniques in code design Use machine resources in concurrent execution Enable programs to work with each other using Message Passing Avoid classic synchronization problems In Detail Creating multithreaded applications is a present-day approach towards programming. With the power of C++, you can easily create various types of applications and perform parallelism and optimizations in your existing work. This book is a practical, powerful, and easy-to-understand guide to C++ multithreading. You will learn how to benefit from the multithreaded approach and enhance your development skills to build better applications. This book will not only help you avoid problems when creating parallel code, but also help you to understand synchronization techniques. The end goal of the book will be to impart various multithreading concepts that will enable you to do parallel computing and concurrent programming quickly and efficiently. Approach The book is an easy-to-follow guide for creating multi-threaded applications using C++. Each topic is thoroughly explained with multiple illustrations. Many algorithms, such as Dining Philosophers Problem give you thorough explanations that will help you to understand and solve concurrent tasks. Who this book is for The book is intended for enterprise developers and programmers who wish to make use of C++ capabilities to learn the multithreaded

approach. Knowledge of multithreading along with experience in C++ is an added advantage. However it is not a prerequisite.

Looks at the principles and clean code, includes case studies showcasing the practices of writing clean code, and contains a list of heuristics and "smells" accumulated from the process of writing clean code.

Presents a collection of tips for programmers on how to use the features of C++11 and C++14 effectively, covering such topics as functions, rvalue references, and lambda expressions.

Concurrent programming has become a required discipline for all programmers. Multi-core processors and the increasing demand for maximum performance and scalability in mission-critical applications have renewed interest in functional languages like Erlang that are designed to handle concurrent programming. Erlang, and the OTP platform, make it possible to deliver more robust applications that satisfy rigorous uptime and performance requirements. Erlang and OTP in Action teaches you to apply Erlang's message passing model for concurrent programming--a completely different way of tackling the problem of parallel programming from the more common multi-threaded approach. This book walks you through the practical considerations and steps of building systems in Erlang and integrating them with real-world C/C++, Java, and .NET applications. Unlike other books on the market, Erlang and OTP in Action offers a comprehensive view of how concurrency relates to SOA and web technologies. This hands-on guide is perfect for readers just learning Erlang or for those who want to apply their theoretical knowledge of this powerful language. You'll delve into the Erlang language and OTP runtime by building several progressively more interesting real-world distributed applications. Once you are competent in the fundamentals of Erlang, the book takes you on a deep dive into the process of designing complex software systems in Erlang. Purchase of the print book comes with an offer of a free PDF, ePub, and Kindle eBook from Manning. Also available is all code from the book.

Templates are among the most powerful features of C++, but they remain misunderstood and underutilized, even as the C++ language and development community have advanced. In C++ Templates, Second Edition, three pioneering C++ experts show why, when, and how to use modern templates to build software that's cleaner, faster, more efficient, and easier to maintain. Now extensively updated for the C++11, C++14, and C++17 standards, this new edition presents state-of-the-art techniques for a wider spectrum of applications. The authors provide authoritative explanations of all new language features that either improve templates or interact with them, including variadic templates, generic lambdas, class template argument deduction, compile-time if, forwarding references, and user-defined literals. They also deeply delve into fundamental language concepts (like value categories) and fully cover all standard type traits. The book starts with an insightful tutorial on basic concepts and relevant language features. The remainder of the book serves as a comprehensive reference, focusing first on language details and then on coding techniques, advanced applications, and sophisticated idioms. Throughout, examples clearly illustrate abstract concepts and demonstrate best practices for exploiting all that C++ templates can do. Understand exactly how templates behave, and avoid common pitfalls Use templates to write more efficient, flexible, and maintainable software Master today's most effective idioms and techniques Reuse source code without compromising performance or safety Benefit from utilities for generic programming in the C++ Standard Library Preview the upcoming concepts feature The companion website, tmplbook.com, contains sample code and additional updates.

An Introduction to Programming by the Inventor of C++ Preparation for Programming in the Real World The book assumes that you aim eventually to write non-trivial programs, whether for work in software development or in some other technical field. Focus on Fundamental Concepts and Techniques The book explains fundamental concepts and techniques in greater depth than traditional introductions. This approach will give you a solid foundation for writing useful, correct, maintainable, and efficient code. Programming with Today's C++ (C++11 and C++14) The book is an introduction to programming in general, including object-oriented programming and generic programming. It is also a solid introduction to the C++ programming language, one of the most widely used languages for real-world software. The book presents modern C++ programming techniques from the start, introducing the C++ standard library and C++11 and C++14 features to simplify programming tasks. For Beginners--And Anyone Who Wants to Learn Something New The book is primarily designed for people who have never programmed before, and it has been tested with many thousands of first-year university students. It has also been extensively used for self-study. Also, practitioners and advanced students have gained new insight and guidance by seeing how a master approaches the elements of his art. Provides a Broad View The first half of the book covers a wide range of essential concepts, design and programming techniques, language features, and libraries. Those will enable you to write programs involving input, output, computation, and simple graphics. The second half explores more specialized topics (such as text processing, testing, and the C programming language) and provides abundant reference material. Source code and support supplements are available from the author's website.

A fast-paced, thorough introduction to modern C++ written for experienced programmers. After reading C++ Crash Course, you'll be proficient in the core language concepts, the C++ Standard Library, and the Boost Libraries. C++ is one of the most widely used languages for real-world software. In the hands of a knowledgeable programmer, C++ can produce small, efficient, and readable code that any programmer would be proud of. Designed for intermediate to advanced programmers, C++ Crash Course cuts through the weeds to get you straight to the core of C++17, the most modern revision of the ISO standard. Part 1 covers the core of the C++ language, where you'll learn about everything from types and functions, to the object life cycle and expressions. Part 2 introduces you to the C++ Standard Library and Boost Libraries, where you'll learn about all of the high-quality, fully-featured facilities available to you. You'll cover special utility classes, data structures, and algorithms, and learn how to manipulate file systems and build high-performance programs that communicate over networks. You'll learn all the major features of modern C++, including:

- Fundamental types, reference types, and user-defined types
- The object lifecycle including storage duration, memory management, exceptions, call stacks, and the RAII paradigm
- Compile-time polymorphism with templates and run-time polymorphism with virtual classes
- Advanced expressions, statements, and functions
- Smart pointers, data structures, dates and times, numerics, and probability/statistics facilities
- Containers, iterators, strings, and algorithms
- Streams and files, concurrency,

networking, and application development With well over 500 code samples and nearly 100 exercises, C++ Crash Course is sure to help you build a strong C++ foundation.

C++17 is a major update to the language and brings many exciting additions and improvements that will change your pre for the better. This book shows you all of the significant changes in the new Standard. I spent hundreds of hours investigating how the new features work to ensure this book is helpful and practical. It will not only save you time but will guide you through lots of nuances of the language and the Standard Library. Among the dozens of C++17's enhancements described in the book, you will learn about the following: Class Template Argument Deduction Structured Bindings Fold Expressions Inline Variables Parallel Algorithms Polymorphic Allocator `std::any` `std::byte` `std::filesystem` `std::from_chars` `std::optional` `std::string_view` `std::variant` `[[nodiscard]]` `if constexpr` If you have experience with C++11/14 and you want to advance to the latest C++ Standard, then pick up the book and start reading. "If you've ever asked 'what's in C++17 and what does it mean for me and my pre?' -- and I hope you have -- then this book is for you." Herb Sutter The book is also available as ebook @Leanpub: <https://leanpub.com/cpp17indetail> C++ Concurrency in Action Practical Multithreading Manning Publications

Program 3D Games in C++: The #1 Language at Top Game Studios Worldwide C++ remains the key language at many leading game development studios. Since it's used throughout their enormous code bases, studios use it to maintain and improve their games, and look for it constantly when hiring new developers. Game Programming in C++ is a practical, hands-on approach to programming 3D video games in C++. Modeled on Sanjay Madhav's game programming courses at USC, it's fun, easy, practical, hands-on, and complete. Step by step, you'll learn to use C++ in all facets of real-world game programming, including 2D and 3D graphics, physics, AI, audio, user interfaces, and much more. You'll hone real-world skills through practical exercises, and deepen your expertise through start-to-finish projects that grow in complexity as you build your skills. Throughout, Madhav pays special attention to demystifying the math that all professional game developers need to know. Set up your C++ development tools quickly, and get started Implement basic 2D graphics, game updates, vectors, and game physics Build more intelligent games with widely used AI algorithms Implement 3D graphics with OpenGL, shaders, matrices, and transformations Integrate and mix audio, including 3D positional audio Detect collisions of objects in a 3D environment Efficiently respond to player input Build user interfaces, including Head-Up Displays (HUDs) Improve graphics quality with anisotropic filtering and deferred shading Load and save levels and binary game data Whether you're a working developer or a student with prior knowledge of C++ and data structures, Game Programming in C++ will prepare you to solve real problems with C++ in roles throughout the game development lifecycle. You'll master the language that top studios are hiring for—and that's a proven route to success.

Threads are a fundamental part of the Java platform. As multicore processors become the norm, using concurrency effectively becomes essential for building high-performance applications. Java SE 5 and 6 are a huge step forward for the development of concurrent applications, with improvements to the Java Virtual Machine to support high-performance, highly scalable concurrent classes and a rich set of new concurrency building blocks. In Java Concurrency in Practice , the creators of these new facilities explain not only how they work and how to use them, but also the motivation and design patterns behind them. However, developing, testing, and debugging multithreaded programs can still be very difficult; it is all too easy to create concurrent programs that appear to work, but fail when it matters most: in production, under heavy load. Java Concurrency in Practice arms readers with both the theoretical underpinnings and concrete techniques for building reliable, scalable, maintainable concurrent applications. Rather than simply offering an inventory of concurrency APIs and mechanisms, it provides design rules, patterns, and mental models that make it easier to build concurrent programs that are both correct and performant. This book covers: Basic concepts of concurrency and thread safety Techniques for building and composing thread-safe classes Using the concurrency building blocks in `java.util.concurrent` Performance optimization dos and don'ts Testing concurrent programs Advanced topics such as atomic variables, nonblocking algorithms, and the Java Memory Model

Summary Functional Programming in C++ teaches developers the practical side of functional programming and the tools that C++ provides to develop software in the functional style. This in-depth guide is full of useful diagrams that help you understand FP concepts and begin to think functionally. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology Well-written code is easier to test and reuse, simpler to parallelize, and less error prone. Mastering the functional style of programming can help you tackle the demands of modern apps and will lead to simpler expression of complex program logic, graceful error handling, and elegant concurrency. C++ supports FP with templates, lambdas, and other core language features, along with many parts of the STL. About the Book Functional Programming in C++ helps you unleash the functional side of your brain, as you gain a powerful new perspective on C++ coding. You'll discover dozens of examples, diagrams, and illustrations that break down the functional concepts you can apply in C++, including lazy evaluation, function objects and invocables, algebraic data types, and more. As you read, you'll match FP techniques with practical scenarios where they offer the most benefit. What's inside Writing safer code with no performance penalties Explicitly handling errors through the type system Extending C++ with new control structures Composing tasks with DSLs About the Reader Written for developers with two or more years of experience coding in C++. About the Author Ivan ?uki? is a core developer at KDE and has been coding in C++ since 1998. He teaches modern C++ and functional programming at the Faculty of Mathematics at the University of Belgrade. Table of Contents Introduction to functional programming Getting started with functional programming Function objects Creating new functions from the old ones Purity: Avoiding mutable state Lazy evaluation Ranges Functional data structures Algebraic data types and pattern matching Monads Template metaprogramming Functional design for concurrent systems Testing and debugging

Go from competent C++ developer to skilled designer or architect using this book as your C++ design master class. This title will guide you through the design and implementation of a fun, engaging case study. Starting with a quick exploration of the requirements for building the application, you'll delve into selecting an appropriate architecture, eventually designing and implementing all of the necessary modules to meet the project's requirements. By the conclusion of Practical C++ Design, you'll have constructed a fully functioning calculator that builds and executes on multiple platforms. Access to the complete source code will help speed your learning. Utilize the Model-View-Controller pattern to determine the optimal architecture for the calculator; the observer pattern to design an event system; the singleton pattern as you design the calculator's central data repository, a reusable stack; the command pattern to design a command system supporting unlimited undo/redo; and the abstract factory pattern for a cross-platform plugin infrastructure to make the calculator extensible. What You Will Learn Read a specification document and translate it into a practical C++ design Understand trade-offs in selecting between alternative design scenarios Gain

practical experience in applying design patterns to realistic development scenarios Learn how to effectively use language elements of modern C++ to create a lasting design Develop a complete C++ program from a blank canvas through to a fully functioning, cross platform application Read, modify, and extend existing, high quality code Learn the fundamentals of API design, including class, module, and plugin interfaces Who This Book Is For The experienced C++ developer ready to take the next step to becoming a skilled C++ designer.

[Copyright: 75c40c12553c0f952fd75147fe41c7d9](#)