

## More Effective C Scott Meyers

C++ is a complex language with many subtle facets. This is especially true when it comes to object-oriented and template programming. The C++ Pocket Reference is a memory aid for C++ programmers, enabling them to quickly look up usage and syntax for unfamiliar and infrequently used aspects of the language. The book's small size makes it easy to carry about, ensuring that it will always be at-hand when needed. Programmers will also appreciate the book's brevity; as much information as possible has been crammed into its small pages. In the C++ Pocket Reference, you will find: Information on C++ types and type conversions Syntax for C++ statements and preprocessor directives Help declaring and defining classes, and managing inheritance Information on declarations, storage classes, arrays, pointers, strings, and expressions Refreshers on key concepts of C++ such as namespaces and scope More! C++ Pocket Reference is useful to Java and C programmers making the transition to C++, or who find themselves occasionally programming in C++. The three languages are often confusingly similar. This book enables programmers familiar with C or Java to quickly come up to speed on how a particular construct or concept is implemented in C++. Together with its companion STL Pocket Reference, the C++ Pocket Reference forms one of the most concise, easily-carried, quick-references to the C++ language available.

This is the eBook version of the printed book. C++'s Standard Template Library is revolutionary, but learning to use it well has always been a challenge. Until now. In this book, best-selling author Scott Meyers (Effective C++, and More Effective C++) reveals the critical rules of thumb employed by the experts - the things they almost always do or almost always avoid doing - to get the most out of the library. Other books describe what's in the STL. Effective STL shows you . Each of the book's 50 guidelines is backed by Meyers' legendary analysis and incisive examples, so you'll learn not o.

Become an accomplished Ruby programmer by understanding the design principles, best practices, and trade-offs involved in implementation approaches to keep your Ruby applications maintainable in the long term Key Features Understand the design principles behind polished Ruby code and trade-offs between implementation approaches Use metaprogramming and DSLs to reduce the amount of code needed without decreasing maintainability Learn Ruby web application design principles and strategies for databases, security, and testing Book Description Most successful Ruby applications become difficult to maintain over time as the codebase grows in size. Polished Ruby Programming provides you with recommendations and advice for designing Ruby programs that are easy to maintain in the long term. This book takes you through implementation approaches for many common programming situations, the trade-offs inherent in each approach, and why you may choose to use different approaches in different situations. You'll start by learning fundamental Ruby programming principles, such as correctly using core classes, class and method design, variable usage, error handling, and code formatting. Moving on, you'll learn higher-level programming principles, such as library design, use of metaprogramming and domain-specific languages, and refactoring. Finally, you'll learn principles specific to web application development, such as how to choose a database and web framework, and how to use advanced security features. By the end of this Ruby programming book, you'll have gained the skills you need to design robust, high-performance, scalable, and maintainable Ruby applications. While most code examples and principles discussed in the book apply to all Ruby versions, some examples and principles are specific to Ruby 3.0, the latest release at the time of publication. What you will learn Use Ruby's core classes and design custom classes effectively Explore the principles behind variable usage and method argument choice Implement advanced error handling approaches such as exponential backoff Design extensible libraries and plugin systems in Ruby Use metaprogramming and DSLs to avoid code redundancy Implement different approaches to testing and understand their trade-offs Discover design patterns, refactoring, and optimization with Ruby Explore database design principles and advanced web app security Who this book is for If you already know how to program in Ruby and want to learn more about the principles and best practices behind writing maintainable, scalable, optimized, and well-structured Ruby code, then this Ruby book is for you. Intermediate to advanced-level working knowledge of the Ruby programming language is expected to get the most out of this book.

OpenGL® SuperBible, Fifth Edition is the definitive programmer's guide, tutorial, and reference for the world's leading 3D API for real-time computer graphics, OpenGL 3.3. The best all-around introduction to OpenGL for developers at all levels of experience, it clearly explains both the API and essential associated programming concepts. Readers will find up-to-date, hands-on guidance on all facets of modern OpenGL development, including transformations, texture mapping, shaders, advanced buffers, geometry management, and much more. Fully revised to reflect ARB's latest official specification (3.3), this edition also contains a new start-to-finish tutorial on OpenGL for the iPhone, iPod touch, and iPad. Coverage includes A practical introduction to the essentials of real-time 3D graphics Core OpenGL 3.3 techniques for rendering, transformations, and texturing Writing your own shaders, with examples to get you started Cross-platform OpenGL: Windows (including Windows 7), Mac OS X, GNU/Linux, UNIX, and embedded systems OpenGL programming for iPhone, iPod touch, and iPad: step-by-step guidance and complete example programs Advanced buffer techniques, including full-definition rendering with floating point buffers and textures Fragment operations: controlling the end of the graphics pipeline Advanced shader usage and geometry management A fully updated API reference, now based on the official ARB (Core) OpenGL 3.3 manual pages New bonus materials and sample code on a companion Web site, [www.starstonesoftware.com/OpenGL](http://www.starstonesoftware.com/OpenGL) Part of the OpenGL Technical Library—The official knowledge resource for OpenGL developers The OpenGL Technical Library provides tutorial and reference books for OpenGL. The Library enables programmers to gain a practical understanding of OpenGL and shows them how to unlock its full potential.

Originally developed by SGI, the Library continues to evolve under the auspices of the OpenGL Architecture Review Board (ARB) Steering Group (now part of the Khronos Group), an industry consortium responsible for guiding the evolution of OpenGL and related technologies.

Scott Meyers's seminal C++ books— Effective C++ , More Effective C++ , and Effective STL —have been immensely helpful to hundreds of thousands of C++ programmers. All three are finally available together in this eBook collection. Effective C++ has been embraced by hundreds of thousands of programmers worldwide. The reason is clear: Scott Meyers's practical approach to C++ describes the rules of thumb used by the experts to produce clear, correct, efficient code. The book is organized around 55 specific guidelines, each of which describes a way to write better C++. Each is backed by concrete examples. In More Effective C++, Meyers presents 35 ways to improve your programs and designs. Drawing on years of experience, Meyers explains how to write software that is more effective: more efficient, more robust, more consistent, more portable, and more reusable. In short, how to write C++ software that's just plain better. In Effective STL, Meyers goes beyond describing what's in the STL to show you how to use it. Each of the book's 50 guidelines is backed by Meyers's legendary analysis and incisive examples, so you'll learn not only what to do, but also when to do it—and why. Together in this collection, these books include the following important features: Expert guidance on the design of effective classes, functions, templates, and inheritance hierarchies. Applications of new "TR1" standard library functionality, along with comparisons to existing standard library components. Insights into differences between C++ and other languages (e.g., Java, C#, C) that help developers from those languages assimilate "the C++ way" of doing things. Proven methods for improving program efficiency, including incisive examinations of the time/space costs of C++ language features Comprehensive descriptions of advanced techniques used by C++ experts, including placement new, virtual constructors, smart pointers, reference counting, proxy classes, and double-dispatching Examples of the profound impact of exception handling on the structure and behavior of C++ classes and functions Practical treatments of new language features, including bool, mutable, explicit, namespaces, member templates, the Standard Template Library, and more. If your compilers don't yet support these features, Meyers shows you how to get the job done without them. Advice on choosing among standard STL containers (like vector and list), nonstandard STL containers (like hash\_set and hash\_map), and non-STL containers (like bitset). Techniques to maximize the efficiency of the STL and the programs that use it. Insights into the behavior of iterators, function objects, and allocators, including things you should not do. Guidance for the proper use of algorithms and member functions whose names are the same (e.g., find), but whose actions differ in subtle (but important) ways. Discussions of potential portability problems, including straightforward ways to avoid them.

This book gives a good start and complete introduction for C# Programming for Beginner's. While reading this book it is fun and easy to read it. This book is best suitable for first time C# readers, Covers all fast track topics of C# for all Computer Science students and Professionals. This book is targeted toward those who have little or no programming experience or who might be picking up C# as a second language. The book has been structured and written with a purpose: to get you productive as quickly as possible. I've used my experiences in writing applications with C# and teaching C# to create a book that I hope cuts through the fluff and teaches you what you need to know. All too often, authors fall into the trap of focusing on the technology rather than on the practical application of the technology. I've worked hard to keep this book focused on teaching you practical skills that you can apply immediately toward a development project. This book is divided into ten Chapters, each of which focuses on a different aspect of developing applications with C#. These parts generally follow the flow of tasks you'll perform as you begin creating your own programs with C#. I recommend that you read them in the order in which they appear. Using C#, this book develops the concepts and theory of Building the Program Logic and Interfaces analysis, Exceptions, Delegates and Events and other important things in a gradual, step-by-step manner, proceeding from concrete examples to abstract principles. Standish covers a wide range of both traditional and contemporary software engineering topics. This is a handy guide of sorts for any computer science engineering Students, Thinking In C# Programming is a solution bank for various complex problems related to C# and .NET. It can be used as a reference manual by Computer Science Engineering students. This Book also covers all aspects of B.TECH CS, IT, and BCA and MCA, BSC IT. Preview introduced programmers to a new era called functional programming. C# focused on bridging the gap between programming languages and databases. This book covers all the language features from the first version through C# . It also provides you with the essentials of using Visual Studio 2005 to let you enjoy its capabilities and save you time by using features such as IntelliSense. Learning a new programming language can be intimidating. If you've never programmed before, the act of typing seemingly cryptic text to produce sleek and powerful applications probably seems like a black art, and you might wonder how you'll ever learn everything you need to know. The answer is, of course, one step at a time. The first step to learning a language is the same as that of any other activity: building confidence. Programming is part art and part science. Although it might seem like magic, it's more akin to illusion: After you know how things work a lot of the mysticism goes away, freeing you to focus on the mechanics necessary to produce any given desired result. Chapter 1 (Introduction To C# AND .NET) Chapter 2 (Your First Go at C# Programming) Chapter 3 (C# Data Types)' Chapter 4 (Building the Program Logic) Chapter 5 (Using Classes) Chapter 6 (Function Members) Chapter 7 (Structs, Enums, and Attributes) Chapter 8 (Interfaces) Chapter 9 (Exceptions) Chapter 10 (Delegates and Events)

Take your C++ coding to the next level by leveraging the latest features and advanced techniques to building high performing, reliable applications. About This Book Get acquainted with the latest features in C++ 17 Take advantage of the myriad of features and possibilities that C++ offers to build real-world applications Write clear and expressive code in C++, and get insights into how to keep your code error-free Who This Book Is For This book is for experienced C++ developers. If you are a novice C++ developer, then

it's highly recommended that you get a solid understanding of the C++ language before reading this book What You Will Learn Write modular C++ applications in terms of the existing and newly introduced features Identify code-smells, clean up, and refactor legacy C++ applications Leverage the possibilities provided by Cucumber and Google Test/Mock to automate test cases Test frameworks with C++ Get acquainted with the new C++17 features Develop GUI applications in C++ Build portable cross-platform applications using standard C++ features In Detail C++ has come a long way and has now been adopted in several contexts. Its key strengths are its software infrastructure and resource-constrained applications. The C++ 17 release will change the way developers write code, and this book will help you master your developing skills with C++. With real-world, practical examples explaining each concept, the book will begin by introducing you to the latest features in C++ 17. It encourages clean code practices in C++ in general, and demonstrates the GUI app-development options in C++. You'll get tips on avoiding memory leaks using smart-pointers. Next, you'll see how multi-threaded programming can help you achieve concurrency in your applications. Moving on, you'll get an in-depth understanding of the C++ Standard Template Library. We show you the concepts of implementing TDD and BDD in your C++ programs, and explore template-based generic programming, giving you the expertise to build powerful applications. Finally, we'll round up with debugging techniques and best practices. By the end of the book, you'll have an in-depth understanding of the language and its various facets. Style and approach This straightforward guide will help you level up your skills in C++ programming, be it for enterprise software or for low-latency applications like games. Filled with real-world, practical examples, this book will take you gradually up the steep learning curve that is C++.

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.

Are you looking for a deeper understanding of the Java™ programming language so that you can write code that is clearer, more correct, more robust, and more reusable? Look no further! *Effective Java™, Second Edition*, brings together seventy-eight indispensable programmer's rules of thumb: working, best-practice solutions for the programming challenges you encounter every day. This highly anticipated new edition of the classic, Jolt Award-winning work has been thoroughly updated to cover Java SE 5 and Java SE 6 features introduced since the first edition. Bloch explores new design patterns and language idioms, showing you how to make the most of features ranging from generics to enums, annotations to autoboxing. Each chapter in the book consists of several "items" presented in the form of a short, standalone essay that provides specific advice, insight into Java platform subtleties, and outstanding code examples. The comprehensive descriptions and explanations for each item illuminate what to do, what not to do, and why. Highlights include: New coverage of generics, enums, annotations, autoboxing, the for-each loop, varargs, concurrency utilities, and much more Updated techniques and best practices on classic topics, including objects, classes, libraries, methods, and serialization How to avoid the traps and pitfalls of commonly misunderstood subtleties of the language Focus on the language and its most fundamental libraries: `java.lang`, `java.util`, and, to a lesser extent, `java.util.concurrent` and `java.io` Simply put, *Effective Java™, Second Edition*, presents the most practical, authoritative guidelines available for writing efficient, well-designed programs.

More than 150,000 copies in print! Praise for Scott Meyers' first book, *Effective C++*: "I heartily recommend *Effective C++* to anyone who aspires to mastery of C++ at the intermediate level or above." – *The C/C++ User's Journal* From the author of the indispensable *Effective C++*, here are 35 new ways to improve your programs and designs. Drawing on years of experience, Meyers explains how to write software that is more effective: more efficient, more robust, more consistent, more portable, and more reusable. In short, how to write C++ software that's just plain better. *More Effective C++* includes: Proven methods for improving program efficiency, including incisive examinations of the time/space costs of C++ language features Comprehensive descriptions of advanced techniques used by C++ experts, including placement new, virtual constructors, smart pointers, reference counting, proxy classes, and double-dispatching Examples of the profound impact of exception handling on the structure and behavior of C++ classes and functions Practical treatments of new language features, including `bool`, `mutable`, `explicit`, namespaces, member templates, the Standard Template Library, and more. If your compilers don't yet support these features, Meyers shows you how to get the job done without them. *More Effective C++* is filled with pragmatic, down-to-earth advice you'll use every day. Like *Effective C++* before it, *More Effective C++* is essential reading for anyone working with C++.

From contributors to animated films such as *Toy Story* and *A Bug's Life*, comes this text to help animators create the sophisticated computer-generated special effects seen in such features as *Jurassic Park*. You're smart and savvy, but also busy. This comprehensive guide to Apple's Mac OS X 10.6, Snow Leopard, gives you everything you need to know to live a happy, productive Mac life. Learn Mac OS X Snow Leopard will have you up and connected lickity-split. With a minimum of overhead and a maximum of useful information, you'll cover a lot of ground in the time it takes other books to get you plugged in. If this isn't your first experience with Mac OS X, skip right to the "What's New in Snow Leopard" sections. You may also find yourself using this book as a quick refresher course or a way to learn new Mac skills you've never tried before.

“Every C++ professional needs a copy of Effective C++. It is an absolute must-read for anyone thinking of doing serious C++ development. If you’ve never read Effective C++ and you think you know everything about C++, think again.” — Steve Schirripa, Software Engineer, Google “C++ and the C++ community have grown up in the last fifteen years, and the third edition of Effective C++ reflects this. The clear and precise style of the book is evidence of Scott’s deep insight and distinctive ability to impart knowledge.” — Gerhard Kreuzer, Research and Development Engineer, Siemens AG The first two editions of Effective C++ were embraced by hundreds of thousands of programmers worldwide. The reason is clear: Scott Meyers’ practical approach to C++ describes the rules of thumb used by the experts — the things they almost always do or almost always avoid doing — to produce clear, correct, efficient code. The book is organized around 55 specific guidelines, each of which describes a way to write better C++. Each is backed by concrete examples. For this third edition, more than half the content is new, including added chapters on managing resources and using templates. Topics from the second edition have been extensively revised to reflect modern design considerations, including exceptions, design patterns, and multithreading. Important features of Effective C++ include: Expert guidance on the design of effective classes, functions, templates, and inheritance hierarchies. Applications of new “TR1” standard library functionality, along with comparisons to existing standard library components. Insights into differences between C++ and other languages (e.g., Java, C#, C) that help developers from those languages assimilate “the C++ way” of doing things.

An exploration of why people all over the world love to engage in pain on purpose—from dominatrices, religious ascetics, and ultramarathoners to ballerinas, icy ocean bathers, and sideshow performers Masochism is sexy, human, reviled, worshipped, and can be delightfully bizarre. Deliberate and consensual pain has been with us for millennia, encompassing everyone from Black Plague flagellants to ballerinas dancing on broken bones to competitive eaters choking down hot peppers while they cry. Masochism is a part of us. It lives inside workaholics, tattoo enthusiasts, and all manner of garden variety pain-seekers. At its core, masochism is about feeling bad, then better—a phenomenon that is long overdue for a heartfelt and hilarious investigation. And Leigh Cowart would know: they are not just a researcher and science writer—they’re an inveterate, high-sensation seeking masochist. And they have a few questions: Why do people engage in masochism? What are the benefits and the costs? And what does masochism have to say about the human experience? By participating in many of these activities themselves, and through conversations with psychologists, fellow scientists, and people who seek pain for pleasure, Cowart unveils how our minds and bodies find meaning and relief in pain—a quirk in our programming that drives discipline and innovation even as it threatens to swallow us whole.

Write Truly Great iOS and OS X Code with Objective-C 2.0! Effective Objective-C 2.0 will help you harness all of Objective-C’s expressive power to write OS X or iOS code that works superbly well in production environments. Using the concise, scenario-driven style pioneered in Scott Meyers’ best-selling Effective C++, Matt Galloway brings together 52 Objective-C best practices, tips, shortcuts, and realistic code examples that are available nowhere else. Through real-world examples, Galloway uncovers little-known Objective-C quirks, pitfalls, and intricacies that powerfully impact code behavior and performance. You’ll learn how to choose the most efficient and effective way to accomplish key tasks when multiple options exist, and how to write code that’s easier to understand, maintain, and improve. Galloway goes far beyond the core language, helping you integrate and leverage key Foundation framework classes and modern system libraries, such as Grand Central Dispatch. Coverage includes Optimizing interactions and relationships between Objective-C objects Mastering interface and API design: writing classes that feel “right at home” Using protocols and categories to write maintainable, bug-resistant code Avoiding memory leaks that can still occur even with Automatic Reference Counting (ARC) Writing modular, powerful code with Blocks and Grand Central Dispatch Leveraging differences between Objective-C protocols and multiple inheritance in other languages Improving code by more effectively using arrays, dictionaries, and sets Uncovering surprising power in the Cocoa and Cocoa Touch frameworks

C++ Sale price. You will save 66% with this offer. Please hurry up! Effective Modern C++(C++ 11, C++ 14) If you are a programmer or looking to get into programming, you are probably wondering what C++11 and C++ 14 have to offer. You're probably wondering about their major differences and ultimately what it can do to help you code more effectively. This book is here to provide that information. C++11 and C++14 have made significant changes to improve not only a variety of libraries but also the core language. C++14 is the newest version of C++ which was released in August of 2014. Improvements in this version made the language not only convenient to use but also safer. This guide will provide more than just information. This guide will provide information on how the language has changed, how you can use it and examples of putting it all together in practice. This book will also provide details various problems and how to solve them from a C++11 and C++14 perspective. Use this book as your reference guide for some of the major features within C++11 and C++14. Here is a preview of what you'll learn: Multithreading support Generic programming support Uniform initialization Performance C++ Standard Library Download your copy of "C++" by scrolling up and clicking "Buy Now With 1-Click" button. Tags: C Programming, C++programming, C++ programming language, HTML, Javascript, Programming, Developers, Coding, CSS, Java, PHP, C++, Javascript, PHP, Python, Sql, HTML, Swift, C++, C Programming, Programming for beginners, c plus plus, PHP, Java, C++ Programming for Beginners, c primer plus, C Programming for Beginners, C++, C Programming, Programming for beginners, c plus plus, PHP, Java, C++ Programming for Beginners, C Programming, C++programming, C++ programming language, HTML, Javascript, Programming, Developers, Coding, CSS, Java, PHP, hackers, hacking, how to hack, hacking exposed, hacking system, hacking 101, hacking for dummies, Hacking Guide, Hacking Essentials, Computer Bugs, Security Breach, internet skills, hacking techniques, computer hacking, hacking the system, web hacking, how to hack

"This is Effective C++ volume three - it's really that good." - Herb Sutter, independent consultant and secretary of the ISO/ANSI C++ standards committee "There are very few books which all C++ programmers must have. Add Effective STL to that list." - Thomas Becker, Senior Software Engineer, Zephyr Associates, Inc., and columnist, C/C++ Users Journal C++'s Standard Template Library is revolutionary, but learning to use it well has always been a challenge. Until now. In this book, best-selling author Scott Meyers ( Effective C++ , and More Effective C++ ) reveals the critical rules of thumb employed by the experts - the things they almost always do or almost always avoid doing - to get the most out of the library. Other books describe what's in the STL. Effective STL shows you how to use it. Each of the book's 50 guidelines is backed by Meyers' legendary analysis and incisive examples, so you'll learn not only what to do, but also when to do it - and why. Highlights of Effective STL include: Advice on choosing among standard STL containers (like vector and list), nonstandard STL containers (like hash\_set and hash\_map), and non-STL containers (like bitset). Techniques to maximize the efficiency of the STL and the programs that use it. Insights into the behavior of iterators, function objects, and allocators, including things you should not do. Guidance for the proper use of algorithms and member functions whose names are the same (e.g., find), but whose actions differ in subtle (but important) ways. Discussions of potential portability problems, including straightforward ways to avoid them. Like Meyers' previous books, Effective STL is filled with proven wisdom that comes only from experience. Its clear, concise, penetrating style makes it an essential resource for every STL programmer.

Consistent, high-quality coding standards improve software quality, reduce time-to-market, promote teamwork, eliminate time wasted on inconsequential matters, and simplify

maintenance. Now, two of the world's most respected C++ experts distill the rich collective experience of the global C++ community into a set of coding standards that every developer and development team can understand and use as a basis for their own coding standards. The authors cover virtually every facet of C++ programming: design and coding style, functions, operators, class design, inheritance, construction/destruction, copying, assignment, namespaces, modules, templates, genericity, exceptions, STL containers and algorithms, and more. Each standard is described concisely, with practical examples. From type definition to error handling, this book presents C++ best practices, including some that have only recently been identified and standardized--techniques you may not know even if you've used C++ for years. Along the way, you'll find answers to questions like What's worth standardizing--and what isn't? What are the best ways to code for scalability? What are the elements of a rational error handling policy? How (and why) do you avoid unnecessary initialization, cyclic, and definitional dependencies? When (and how) should you use static and dynamic polymorphism together? How do you practice "safe" overriding? When should you provide a no-fail swap? Why and how should you prevent exceptions from propagating across module boundaries? Why shouldn't you write namespace declarations or directives in a header file? Why should you use STL vector and string instead of arrays? How do you choose the right STL search or sort algorithm? What rules should you follow to ensure type-safe code? Whether you're working alone or with others, C++ Coding Standards will help you write cleaner code--and write it faster, with fewer hassles and less frustration.

Presents a collection of tips for programmers on ways to improve programming skills.

CD-ROM contains: DJGPP compiler and source code for the examples in the book.

What Every Professional C++ Programmer Needs to Know—Pared to Its Essentials So It Can Be Efficiently and Accurately Absorbed C++ is a large, complex language, and learning it is never entirely easy. But some concepts and techniques must be thoroughly mastered if programmers are ever to do professional-quality work. This book cuts through the technical details to reveal what is commonly understood to be absolutely essential. In one slim volume, Steve Dewhurst distills what he and other experienced managers, trainers, and authors have found to be the most critical knowledge required for successful C++ programming. It doesn't matter where or when you first learned C++. Before you take another step, use this book as your guide to make sure you've got it right! This book is for you if You're no "dummy," and you need to get quickly up to speed in intermediate to advanced C++ You've had some experience in C++ programming, but reading intermediate and advanced C++ books is slow-going You've had an introductory C++ course, but you've found that you still can't follow your colleagues when they're describing their C++ designs and code You're an experienced C or Java programmer, but you don't yet have the experience to develop nuanced C++ code and designs You're a C++ expert, and you're looking for an alternative to answering the same questions from your less-experienced colleagues over and over again C++ Common Knowledge covers essential but commonly misunderstood topics in C++ programming and design while filtering out needless complexity in the discussion of each topic. What remains is a clear distillation of the essentials required for production C++ programming, presented in the author's trademark incisive, engaging style.

Mac OS X Leopard: Beyond the Manual is written for the sophisticated computer user who would find an introductory manual tedious. Features new to Leopard are emphasized, as are complex features that, though available in earlier versions of OS X, were not readily accessible. The narrative is fast-paced, concise, and respectful of the reader's familiarity with earlier versions of the program.

Organized in a practical problem-and-solution format, More Exceptional C++ picks up where the widely acclaimed Exceptional C++ leaves off, providing successful strategies for solving real-world problems in C++. Drawing from years of in-the-trenches experience, Herb Sutter provides tested techniques and practical solutions for programmers designing modern software systems with C++, from small projects to enterprise applications. Built around forty programming puzzles, More Exceptional C++ helps you understand the rules and issues critical to successful software design and development in C++. New themes included in this sequel place a strong emphasis on generic programming, memory management, and using the C++ standard library, including coverage of important techniques like traits and predicates. Also included are guidelines and considerations to remember when using standard containers and algorithms--topics rarely covered in-depth in other sources. Readers will find solutions to such important questions as: What pitfalls might you encounter when using `std::map` and `std::set`, and how can you safely avoid them? What kinds of predicates are safe to use with the STL, what kinds aren't, and why? What techniques are available for writing powerful generic template code that can change its own behavior based on the capabilities of the types it's given to work with? When and how should you optimize your code? Why can (and do) fancy optimizations get us into trouble? And how can some of these answers change if you're writing multithread-safe code? Does exception safety affect class design, or can it be retrofitted in as an afterthought? How can you avoid the Siamese Twin problem when combining inheritance-based libraries from different vendors? How can you safely use `auto_ptr`, and then use common design patterns to adapt it to avoid common pitfalls? Can you use `auto_ptr` as a class member? What must you know before you elect to use it that way? Plus one of the most frequently recurring questions about modern C++: When and how should you use namespaces, anyway? A must-have for the serious programmer, More Exceptional C++ provides a thorough and pragmatic understanding of the language while showing you how to write exceptional code in C++. 020170434XB11092001.

Presents a collection of reusable design artifacts, called generic components, together with the techniques that make them possible. The author describes techniques for policy-based design, partial template specialization, typelists, and local classes, then goes on to implement generic components for smart pointers, object factories, functor objects, the

Visitor design pattern, and multimethod engines. c. Book News Inc.

With this book, Christopher Kormanyos delivers a highly practical guide to programming real-time embedded microcontroller systems in C++. It is divided into three parts plus several appendices. Part I provides a foundation for real-time C++ by covering language technologies, including object-oriented methods, template programming and optimization. Next, part II presents detailed descriptions of a variety of C++ components that are widely used in microcontroller programming. It details some of C++'s most powerful language elements, such as class types, templates and the STL, to develop components for microcontroller register access, low-level drivers, custom memory management, embedded containers, multitasking, etc. Finally, part III describes mathematical methods and generic utilities that can be employed to solve recurring problems in real-time C++. The appendices include a brief C++ language tutorial, information on the real-time C++ development environment and instructions for building GNU GCC cross-compilers and a microcontroller circuit. For this third edition, the most recent specification of C++17 in ISO/IEC 14882:2017 is used throughout the text. Several sections on new C++17 functionality have been added, and various others reworked to reflect changes in the standard. Also several new sample projects are introduced and existing ones extended, and various user suggestions have been incorporated. To facilitate portability, no libraries other than those specified in the language standard itself are used. Efficiency is always in focus and numerous examples are backed up with real-time performance measurements and size analyses that quantify the true costs of the code down to the very last byte and microsecond. The target audience of this book mainly consists of students and professionals interested in real-time C++. Readers should be familiar with C or another programming language and will benefit most if they have had some previous experience with microcontroller electronics and the performance and size issues prevalent in embedded systems programming.

This is the eBook version of the printed book. From the author of the indispensable Effective C++, here are 35 new ways to improve your programs and designs. Drawing on years of experience, Meyers explains how to write software that is more effective: more efficient, more robust, more consistent, more portable, and more reusable. In short, how to write C++ software that's just plain better. More Effective C++ includes: Proven methods for improving program efficiency, including incisive examinations of the time/space costs of C++ language features Comprehensive descriptions of advanced techn.

Bjarne Stroustrup's own C++ In-Depth Series is now available all together in one attractive gift box, at a special reduced price! All books in this series have been hand-picked by Bjarne Stroustrup, the creator of the C++ programming language, as being worthy additions to the C++ literature. They give programmers concise, focused guides to specific topics. The series' practical approach is designed to lift professionals to the next level in their programming skills. They are all written by acknowledged experts. The books included are: Modern C++ Design, by Andrei Alexandrescu Accelerated C++, by Andrew Koenig and Barbara Moo Essential C++, by Stan Lippman Exceptional C++, by Herb Sutter More Exceptional C++, by Herb Sutter These are five great books of use to all C++ programmers. They are gathered into one handsome and sturdy gift box, and they are specially priced at over \$30 off the cost of buying them individually. The C++ In-Depth Box Set will be a welcome gift for any C++ programmer. 0201775816B12112002

Effective C++ has been updated to reflect the latest ANSI/ISO standards. The author, a recognised authority on C++, shows readers fifty ways to improve their programs and designs.

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

More Exceptional C++ continues where Herb Sutter's best-selling Exceptional C++ left off, delivering 40 puzzles that illuminate the most challenging -- and most powerful -- aspects of C++. More Exceptional C++ offers many new puzzles focused on generic programming and the C++ Standard Template Library, including important techniques such as traits and predicates, as well as key considerations in using standard containers and algorithms -- many of them never covered elsewhere. More Exceptional C++ contains a detailed new section (and two appendices) on optimization in single- and multithreaded environments. It also provides important new insights on crucial topics first introduced in Exceptional C++, including exception safety, generic programming, and memory management. For all C++ programmers.

"It's easy to start writing code with Python: that's why the language is so immensely popular. However, Python has unique strengths, charms, and expressivity that can be hard to grasp at first -- as well as hidden pitfalls that can easily trip you up if you aren't aware of them. Effective Python will help you harness the full power of Python to write exceptionally robust, efficient, maintainable, and well-performing code. Utilizing the concise, scenario-driven style pioneered in Scott Meyers's best-selling Effective C++, Brett Slatkin brings together 53 Python best practices, tips, shortcuts, and realistic code examples from expert programmers. Through realistic examples, Slatkin uncovers little-known Python quirks, intricacies, and idioms that powerfully impact code behavior and performance. You'll learn how to choose the most efficient and effective way to accomplish key tasks when multiple options exist, and how to write code that's easier to understand, maintain, and improve. Drawing on his deep understanding of Python's capabilities, Slatkin offers practical advice for each major area of development with both Python 3.x and Python 2.x. Coverage includes: \* Algorithms \* Objects \* Concurrency \* Collaboration \* Built-in modules \* Production techniques \* And more Each section contains specific, actionable guidelines organized into items, each with carefully worded advice supported by detailed technical arguments and illuminating examples. Using Effective Python, you can systematically improve all the Python code you write: not by blindly following rules or mimicking incomprehensible idioms, but by gaining a deep understanding of the technical reasons why they make sense."--[Source inconnue].

As Python continues to grow in popularity, projects are becoming larger and more complex. Many Python developers are now taking an interest in high-level software design patterns such as hexagonal/clean architecture, event-driven architecture, and the strategic patterns prescribed by domain-driven design (DDD). But translating those patterns into Python isn't always straightforward. With this hands-on guide, Harry Percival and Bob Gregory from MADE.com introduce proven architectural design patterns to help Python developers manage application complexity—and get the most value out of their test suites. Each

pattern is illustrated with concrete examples in beautiful, idiomatic Python, avoiding some of the verbosity of Java and C# syntax. Patterns include: Dependency inversion and its links to ports and adapters (hexagonal/clean architecture) Domain-driven design's distinction between entities, value objects, and aggregates Repository and Unit of Work patterns for persistent storage Events, commands, and the message bus Command-query responsibility segregation (CQRS) Event-driven architecture and reactive microservices

Bestselling Programming Tutorial and Reference Completely Rewritten for the New C++11 Standard Fully updated and recast for the newly released C++11 standard, this authoritative and comprehensive introduction to C++ will help you to learn the language fast, and to use it in modern, highly effective ways. Highlighting today's best practices, the authors show how to use both the core language and its standard library to write efficient, readable, and powerful code. C++ Primer, Fifth Edition, introduces the C++ standard library from the outset, drawing on its common functions and facilities to help you write useful programs without first having to master every language detail. The book's many examples have been revised to use the new language features and demonstrate how to make the best use of them. This book is a proven tutorial for those new to C++, an authoritative discussion of core C++ concepts and techniques, and a valuable resource for experienced programmers, especially those eager to see C++11 enhancements illuminated. Start Fast and Achieve More Learn how to use the new C++11 language features and the standard library to build robust programs quickly, and get comfortable with high-level programming Learn through examples that illuminate today's best coding styles and program design techniques Understand the "rationale behind the rules": why C++11 works as it does Use the extensive crossreferences to help you connect related concepts and insights Benefit from up-to-date learning aids and exercises that emphasize key points, help you to avoid pitfalls, promote good practices, and reinforce what you've learned Access the source code for the extended examples from [informit.com/title/0321714113](http://informit.com/title/0321714113) C++ Primer, Fifth Edition, features an enhanced, layflat binding, which allows the book to stay open more easily when placed on a flat surface. This special binding method—notable by a small space inside the spine—also increases durability.

The topic is of prime importance to software professionals involved in large development efforts such as databases, operating systems, compilers, and frameworks. This volume explains the process of decomposing large systems into physical (not inheritance) hierarchies of small, manageable components. Concepts and techniques are illustrated with "war stories" from the development firm, Mentor Graphics, as well as with a large-scale example comprising some 12,000 lines of code. Annotation copyright by Book News, Inc., Portland, OR

Meyers provides 50 short, specific, easy-to-remember guidelines that experienced C++ programmers either almost always do or almost always avoid. These rules are each followed by an explanation of the rule's important advice on how to implement it, and are supported by actual programming examples.

Includes the text of Effective C++, Second Edition, and More Effective C++, and a collection of C++ magazine articles. This CD serves as a useful resource for any programmer who wants to learn C++.

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.

[Copyright: 9632a0d8147497b539dd5b2fd3956148](https://www.amazon.com/dp/0321714113)