The Standard C Library

The GNU C LibraryGeneric Programming and the STLProgramming ChallengesC++ Core GuidelinesUsing the C++ Standard Template LibrariesPortable CC++ TemplatesUsing the STLProgramming C# 8.0The Draft Standard C++ LibraryA Tour of C++The C++ Standard Library ExtensionsData Structure ProgrammingThe Python 3 Standard Library by ExampleUnderstanding and Using C PointersThe CERT C Coding StandardMastering the C++17 STLEffective STLPOSIX.4 Programmers GuideStandard C Date/Time LibraryC++ Concurrency in ActionThe Standard C LibraryThe Boost C++ LibrariesThe Practice of ProgrammingModern CC++17 Standard Library Quick ReferenceProgramming in LuaThe C++ Standard LibraryThe C++ Standard LibraryEffective CThe C++ Programming LanguageStandard CThe C++ Standard Template LibraryBeyond the C++ Standard LibraryC Programming FAQsC++: a DialogueC P+ S P+ S PrimerC++ Standard Library Quick ReferenceGnu C Library 2.22 Reference Manual 1/2Python Standard Library

The GNU C Library

"Unlike many beginners' books, C++: A Dialog uses industry-standard C++ and the latest standard libraries - giving you skills you can use with any standard C++ toolset, in any programming environment. You even get all the example code and a standard C++ compiler on CD-ROM so you can write and compile your own standard C++ programs on any 32-bit Microsoft Windows platform."--BOOK JACKET.Title Summary field provided by Blackwell North America, Inc. All Rights Reserved

Generic Programming and the STL

A comprehensive guide to implementing the standard C libraries. Contains advanced functions generally used by people writing larger applications, such as applications interacting with other computers over networks or configuring your system. Topics covered include threads, processes, network communications, signal handling, logging, and system management. Contains code examples and usage recommendations. SALES NOTE: This is the first new book on standard C libraries on the market in some time.

Programming Challenges

C# is undeniably one of the most versatile programming languages available to engineers today. With this comprehensive guide, you’ll learn just how powerful the combination of C# and .NET can be. Author Ian Griffiths guides you through C# 8.0 fundamentals and techniques for building cloud, web, and desktop applications. Designed for experienced programmers, this book provides many code examples to help you work with the nuts and bolts of C#, such as generics, LINQ, and asynchronous programming features. You’ll get up to speed on .NET Core and the latest C# 8.0 additions, including asynchronous streams, nullable references, pattern matching, default interface implementation, ranges and new indexing syntax, and changes in the .NET tool chain. Discover how C# supports fundamental coding features, such as classes, other custom types, collections, and error handling Learn how to write high-performance memory-efficient code with .NET Core’s Span and Memory types Query and process diverse data sources, such as in-memory object models, databases, data streams, and XML documents with LINQ Use .NET’s multithreading features to exploit your computer’s parallel processing capabilities Learn how asynchronous language features can help improve application responsiveness and scalability

C++ Core Guidelines

Written by the originator of the USENET C FAQ, this book addresses the real-world problems on C programming that are asked, again and again, on the "comp.lang.c" newsgroup. The book is aimed at C programmers who need quick, concise answers to the stubborn questions which invariably arise when programming in C. It provides accurate answers, insightful explanations, and extensive code examples.

Using the C++ Standard Template Libraries

Portable C

This is the eBook of the printed book and may not include any media, website access codes, or print supplements that may come packaged with the bound book. Master the Powerful Python 3 Standard Library through Real Code Examples “The genius of Doug’s approach is that with 15 minutes per week, any motivated programmer can learn the Python Standard Library. Doug’s guided tour will help you flip the switch to fully power-up Python’s batteries.” –Raymond Hettinger, Distinguished Python Core Developer The Python 3 Standard Library contains hundreds of modules for interacting with the operating system, interpreter, and Internet–all extensively tested and ready to jump-start application development. Now, Python expert Doug Hellmann introduces every major area of the Python 3.x library through concise source code and output examples. Hellmann’s examples fully demonstrate each feature and are designed for easy learning and reuse. You’ll find practical code for working with text, data structures, algorithms, dates/times, math, the file system, persistence, data exchange, compression, archiving, crypto, processes/threads, networking, Internet capabilities, email, developer and language tools, the runtime, packages, and more. Each section fully covers one module, with links to additional resources, making this book an ideal tutorial and reference. The Python 3 Standard Library by Example introduces Python 3.x’s new libraries, significant functionality changes, and new layout and naming conventions. Hellmann also provides expert porting guidance for moving code from 2.x Python standard library modules to their Python 3.x equivalents. Manipulate text with string, textwrap, re (regular expressions), and difflib Use data structures: enum, collections, array, heapq, queue, struct, copy, and more Implement algorithms elegantly and concisely with functools, itertools, and contextlib Handle dates/times and advanced mathematical tasks Archive and data compression Understand data exchange and persistence, including json, dbm, and sqlite Sign and verify messages cryptographically Manage concurrent operations with processes and threads Test, debug, compile, profile, language, import, and package tools Control interaction at runtime with interpreters or the environment

C++ Templates

"TR1 roughly doubles the size of the C++ standard library, and it introduces many new facilities and even new kinds of library components. TR1 has some classes, for example, where some nested types may or may not exist depending on the template arguments. To programmers whose experience stops with the standard library, this is strange and unfamiliar. This book is complete (it covers all TR1 facilities), it is easier to understand than TR1 itself, and it is technically accurate." --Matthew Austern,software engineer, Google "TR1 will help make the C++ programmer more productive than ever. In this book, Pete Becker has written the ultimate reference guide to these components, what they are, how they work, and what they're used for. This book should be on the bookshelf of anyone who wants to use these standardized components to improve both their productivity as well as their coding quality." --John Maddock, consultant and programmer The current C++ standard library extends the core C++ language with common classes and functions. In recent years, to address limitations in that library, a number of components have been developed to extend the language even further. Compiled in a comprehensive technical report (TR1), the bulk of these extensions have been approved for the next revision of the C++ standard. In this book, Pete Becker describes in detail each component in the TR1 library, explaining new facilities for utilities, containers, call wrappers, type traits, numerics, regular expressions, and C compatibility. He draws on his own experience implementing these components to illustrate their value, clarifying the specifications when necessary and providing complete, tested code examples. Most chapters include exercises of various degrees of difficulty to help programmers get hands-on practice with the new components. Answers to the exercises, along with all code examples, are available on the Web. Appendixes comprise a summary of headers included in or extended by the TR1 library, as well as guidelines on how to use the components safely in multithreaded applications. The C++ Standard Library Extensions is for any programmer who wants to get a jump on the revised standard. It also makes the perfect companion to The C++ Standard Library, by Nicolai Josuttis, both books being tutorials and references essential for using C++ more effectively.

Using the STL

A concise tutorial on making the most of C's portability. Chapters include The C-World Abstract Machine; Portable Use of Functions; Portable Use of Pointers; and more.

Programming C# 8.0

This quick reference is a condensed guide to the essential data structures, algorithms, and functions provided by the C++ Standard Library. Used by millions of C++ programmers on a daily basis, the C++ Standard Library features core classes for strings, I/O streams, and various generic containers, as well as a comprehensive set of algorithms to manipulate them. In recent years, the C++11 and C++14 standards have added even more efficient container classes, a new powerful regular expression library, and a portable multithreading library featuring threads, mutexes, condition variables, and atomic variables. Needless to say, it is hard to know and remember all the possibilities, details, and intricacies of this vast and growing library. This handy reference guide is therefore indispensable to any C++ programmer. It offers a condensed, well-structured summary of all essential aspects of the C++ Standard Library. No page-long, repetitive examples or obscure, rarely used features. Instead, everything you need to know and watch out for in practice is outlined in a compact, to-the-point style, interspersed with practical tips and well-chosen, clarifying examples. The book does not explain the C++ language or syntax, but is accessible to anyone with basic C++ knowledge or programming experience. Even the most experienced C++ programmer though will learn a thing or two from it and find it a useful memory-aid. Among the topics covered are: What You Will Learn Gain the essentials that the C++ Standard Library has to offer Use containers to efficiently store and retrieve your data Use algorithms to inspect and manipulate your data See how lambda expressions allow for elegant use of algorithms Discover what the standard string class provides and how to use it Write localized applications Work with file and stream-based I/O Discover what smart pointers are and how to use them to prevent memory leaks Write safe and efficient multi-threaded code using the threading libraries Who This Book Is For All C++ programmers: irrespective of their proficiency with the language or the Standard Library, this book offers an indispensable reference and memory-aid. A secondary audience is developers who are new to C++, but not new to programming, and who want to learn more on the C++ Standard Library in a quick, condensed manner.

The Draft Standard C++ Library

Written in an informal, informative style, this authoritative guide goes way beyond the standard reference manual. It discusses each of the POSIX.4 facilities and what they mean, why and when you would use each of these facilities, and trouble spots you might run into. c.

A Tour of C++

This new edition reflects C++ in its latest release, 3.0 and its new addition the template facility. Highlights include object-oriented design; exception handling; and the difference between Release 2.0 and 3.0.

The C++ Standard Library Extensions

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.

Data Structure Programming

Authored by Roberto Ierusalimschy, the chief architect of the language, this volume covers all aspects of Lua 5---from the basics to its API with C---explaining how to make good use of its features and giving numerous code examples. (Computer Books)

The Python 3 Standard Library by Example

The GNU C Library, described in this document, defines all of the library functions that are specified by the ISO C standard, as well as additional features specific to POSIX and other derivatives of the Unix operating system, and extensions specific to GNU systems. The purpose of this manual is to tell you how to use the facilities of the GNU C Library. We have mentioned which features belong to which standards to help you identify things that are potentially non-portable to other systems. But the emphasis in this manual is not on strict portability. As the GNU C Library became such a big project over the years, we had to split this reference manual in two parts that are two separate physical books. To keep it consistent with the digital manual, the references and page numbers cover both physical books as it were one. Therefore please note that you probably want to have both parts.

Understanding and Using C Pointers

If you think "Modern" and "C" don't belong in the same sentence, think again. The C standards committee actively reviews and extends the language, with updated published C standards as recently as 2018. In Modern C, author Jens Gustedt teaches you the skills and features you need to write relevant programs in this tried-and-true language, including Linux and Windows, device drivers, web servers and browsers, smartphones, and much more! Modern C teaches you to take your C programming skills to new heights, whether you're just starting out with C or have more extensive experience. Organized by level, this comprehensive guide lets you jump in where it suits you best while still reaping the maximum benefits. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications.

The CERT C Coding Standard

Contains full coverage of the ANSI/ISO C++ standard. The text covers classes, methods, interfaces and objects that make up the standard C++ libraries.

Mastering the C++17 STL

Defines the template classes and functions of the standard template library (STL) component of the C++ programming language. A chapter is devoted to each of the 13 headers, providing a functional description of the header contents, suggestions for how best to use the facilities defined in the header, and the C++ code itself. Additional chapters introduce STL as a whole and discuss three overarching topics--iterators, algorithms, and containers. c. Book News Inc.

Effective STL

A proven best-seller, updated for the new standard: still the most practical C++ Standard Library tutorial and most complete reference * *Thoroughly documents each library component incorporated in the brand-new C++ standard. *Clearly explains complex concepts, and presents the practical detail programmers need to use the Standard Library effectively. *Contains many examples of working code, all available for download at an accompanying website. The C++ Standard Library, 2/e, doesn't just provide comprehensive documentation of every library component associated with the newest C++ standard: it also offers clearly-written explanations of complex concepts, reviews practical programming details needed for effective use, and presents many useful examples of working code - all of them available for download. Fully updated to reflect the newest elements of the C++ Standard Library incorporated into the full ANSI/ISO C++ language standard, this book examines containers, iterators, function objects, STL algorithms, special containers, strings, numerical classes, internationalization, the IOStream library, and much more. Every component is presented in depth: Josuttis explains its purpose and design, presents crystal-clear examples, identifies traps and pitfalls, and offers exact signatures and definitions of its classes and functions. Comprehensive, detailed, readable, and practical, Josuttis' The C++ Standard Library has established itself as the definitive book on the topic: working developers will find this new edition even more useful.

POSIX.4 Programmers Guide

This textbook provides an introduction to data structures and the Standard Template Library (STL), which has been recently accepted by the C++ Standards Committee. It provides a carefully integrated discussion of general data structures together with their implementation and use in the STL, thus teaching readers the important features of abstraction whilst using the STL to develop applications.

Standard C Date/Time Library

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.

C++ Concurrency in Action

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

The Standard C Library

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.

The Boost C++ Libraries

This guide explains the C++ Standard Template Library (STL) in terms of generic programming--a way of designing and writing programs so that they can easily be reused. The fundamental premise is that the STL should be regarded as a library of concepts, rather than as a library of functions and classes.

The Practice of Programming

This quick reference is a condensed guide to the essential data structures, algorithms, and functions provided by the C++17 Standard Library. It does not explain the C++ language or syntax, but is accessible to anyone with basic C++ knowledge or programming experience. Even the most experienced C++ programmer will learn a thing or two from it and find it a useful memory-aid. It is hard to remember all the possibilities, details, and intricacies of the vast and growing Standard Library. This handy reference guide is therefore indispensable to any C++ programmer. It offers a condensed, well-structured summary of all essential aspects of the C++ Standard Library. No page-long, repetitive examples or obscure, rarely used features. Instead, everything you need to know and watch out for in practice is outlined in a compact, to-the-point style, interspersed with practical tips and well-chosen, clarifying examples. This new edition is updated to include all Standard Library changes in C++17, including the new vocabulary types std::string_view, any, optional, and variant; parallel algorithms; the file system library; specialized mathematical functions; and more. What You Will Learn Gain the essentials that the C++ Standard Library has to offer Use containers to efficiently store and retrieve your data Inspect and manipulate your data with algorithms See how lambda expressions allow for elegant use of algorithms Discover what the standard string class provides and how to use it Write localized applications Work with file and stream-based I/O Prevent memory leaks with smart pointers Write safe and efficient multi-threaded code using the threading libraries Who This Book Is For All C++ programmers, irrespective of their proficiency with the language or the Standard Library. A secondary audience is developers who are new to C++, but not new to programming, and who want to learn more about the C++ Standard Library in a quick, condensed manner.

Modern C

The second edition of The Boost C++ Libraries introduces 72 Boost libraries that provide a wide range of useful capabilities. They help you manage memory and process strings more easily. They provide containers and other data structures that go well beyond what the standard library offers. They make it easy to build platform-independent network applications. Simply put, these 72 libraries greatly expand your C++ toolbox. The second edition contains more than 430 examples. All examples are as short as possible, but they are complete, so you can compile and run them as is. They show you what the Boost libraries offer and give you a head start on using the libraries in your own applications. The goal of this book is to increase your efficiency as a C++ developer and to simplify software development with C++. The Boost libraries introduced in this book will help you write less code with fewer bugs and finish projects faster. You code will be more concise and self-explanatory and more easily adapted when requirements change. The second edition is based on the Boost libraries 1.55.0 and 1.56.0 with the latter version having been released in August 2014. The examples are based on C++11 and have been tested with Visual Studio 2013, GCC 4.8 and Clang 3.3 on various platforms. For Boost libraries which were incorporated into the C++11 standard library, differences between Boost and the standard library are highlighted. The Boost libraries are one of the most important and influential open source C++ libraries. Their source code is available under a permissive free software license. Several Boost libraries have been incorporated into the C++11 standard library. The Boost libraries are developed and supported by the Boost community - a worldwide developer community with a strong interest in pushing C++ boundaries further.

C++17 Standard Library Quick Reference

Identifies and explains the syntax, functions, and expressions of the C programming language, and describes how to use its library of utility programs

Programming in Lua

A comprehensive introduction and guide to the STL, pitched at the level of readers already familiar with C++. It presents a thorough overview of the capabilities of the STL, detailed discussions of the use of containers, descriptions of the algorithms and how they may be used, and how the STL may be extended. An appendix provides an alphabetical reference to the entire STL, making this an extremely useful hands-on text for programmers of C++ and students coming to the STL for the first time.

The C++ Standard Library

Improve your programming through a solid understanding of C pointers and memory management. With this practical book, you’ll learn how pointers provide the mechanism to dynamically manipulate memory, enhance support for data structures, and enable access to hardware. Author Richard Reese shows you how to use pointers with arrays, strings, structures, and functions, using memory models throughout the book. Difficult to master, pointers provide C with much flexibility and power—yet few resources are dedicated to this data type. This comprehensive book has the information you need, whether you’re a beginner or an experienced C or C++ programmer or developer. Get an introduction to pointers, including the declaration of different pointer types Learn about dynamic memory allocation, de-allocation, and alternative memory management techniques Use techniques for passing or returning data to and from functions Understand the fundamental aspects of arrays as they relate to pointers Explore the basics of strings and how pointers are used to support them Examine why pointers can be the source of security problems, such as buffer overflow Learn several pointer techniques, such as the use of opaque pointers, bounded pointers and, the restrict keyword

The C++ Standard Library

Lance Latham provides a library of C programming functions that constitute a complete date and time toolkit. He details the Julian calendar and the calendars of most major cultures of the world, and he supplies the historical knowledge necessary to determine the rules of use and the range of problems that programming a solution must address.

Effective C

Introducing the Boost libraries: the next breakthrough in C++ programming Boost takes you far beyond the C++ Standard Library, making C++ programming more elegant, robust, and productive. Now, for the first time, a leading Boost expert systematically introduces the broad set of Boost libraries and teaches best practices for their use. Writing for intermediate-to-advanced C++ developers, Björn Karlsson briefly outlines all 58 Boost libraries, and then presents comprehensive coverage of 12 libraries you're likely to find especially useful. Karlsson's topics range from smart pointers and conversions to containers and data structures, explaining exactly how using each library can improve your code. He offers detailed coverage of higher-order function objects that enable you to write code that is more concise, expressive, and readable. He even takes you "behind the scenes" with Boost, revealing tools and techniques for creating your own generic libraries. Coverage includes Smart pointers that provide automatic lifetime management of objects and simplify resource sharing Consistent, best-practice solutions for performing type conversions and lexical conversions Utility classes that make programming simpler and clearer Flexible container libraries that solve common problems not covered by the C++ Standard Library Powerful support for regular expressions with Boost.Regex Function objects defined at the call site with Boost.Bind and Boost.Lambda More flexible callbacks with Boost.Function Managed signals and slots (a.k.a. the Observer pattern) with Boost.Signals The Boost libraries are proving so useful that many of them are planned for inclusion in the next version of the C++ Standard Library. Get your head start now, with Beyond the C++ Standard Library.

The C++ Programming Language

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.

Standard C

An innovative reference reveals the many capabilites of the Python Standard Library, which is a compilation of commonly used procedures that can be pasted into a Python script, by providing over 300 real-world example scripts. Original. (Intermediate/Advanced)

The C++ Standard Template Library

Beyond the C++ Standard Library

There are many distinct pleasures associated with computer programming. Craftsm- ship has its quiet rewards, the satisfaction that comes from building a useful object and making it work. Excitement arrives with the ?ash of insight that cracks a previously intractable problem. The spiritual quest for elegance can turn the hacker into an artist. Therearepleasuresinparsimony,insqueezingthelastdropofperformanceoutofclever algorithms and tight coding. Thegames,puzzles,andchallengesofproblemsfrominternationalprogrammingc-petitionsareagreatwaytoexperiencethesepleasureswhileimprovingyouralgorithmic and coding skills. This book contains over 100 problems that have appeared in previous programming contests, along with discussions of the theory and ideas necessary to - tack them. Instant online grading for all of these problems is available from two WWW robot judging sites. Combining this book with a judge gives an exciting new way to challenge and improve your programming skills. This book can be used for self-study, for teaching innovative courses in algorithms and programming, and in training for international competition. To the Reader Theproblemsinthisbookhavebeenselectedfromover1,000programmingproblemsat the Universidad de Valladolid online judge, available athttp://online-judge.uva.es.The judgehasruledonwelloveronemillionsubmissionsfrom27,000registeredusersaround the world to date. We have taken only the best of the best, the most fun, exciting, and interesting problems available.

C Programming FAQs

"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.

C++: a Dialogue

With the same insight and authority that made their book The Unix Programming Environment a classic, Brian Kernighan and Rob Pike have written The Practice of Programming to help make individual programmers more effective and productive. The practice of programming is more than just writing code. Programmers must also assess tradeoffs, choose among design alternatives, debug and test, improve performance, and maintain software written by themselves and others. At the same time, they must be concerned with issues like compatibility, robustness, and reliability, while meeting specifications. The Practice of Programming covers all these topics, and more. This book is full of practical advice and real-world examples in C, C++, Java, and a variety of special-purpose languages. It includes chapters on: debugging: finding bugs quickly and methodically testing: guaranteeing that software works correctly and reliably performance: making programs faster and more compact portability: ensuring that programs run everywhere without change design: balancing goals and constraints to decide which algorithms and data structures are best interfaces: using abstraction and information hiding to control the interactions between components style: writing code that works well and is a pleasure to read notation: choosing languages and tools that let the machine do more of the work Kernighan and Pike have distilled years of experience writing programs, teaching, and working with other programmers to create this book. Anyone who writes software will profit from the principles and guidance in The Practice of Programming .

C P+ S P+ S Primer

First comprehensive treatment of ANSI and ISO standards for the C Library. Includes practical advice on using all 15 headers of the Library and covers the concept design and utilization of libraries. Contains complete codes of C Library and is the companion volume to C Programming Language. An independent consultant, author Plauger is one of the world's leading experts on C and the C Library.

C++ Standard Library Quick Reference

For programmers familiar with the C and C++ programming. This book presents the library portion of the draft ANSI/ISO Standard for the programming language C++, and shows how to use all the library classes and functions. An introductory chapter discusses how the Standard C library changes to meet the needs of C++.

Gnu C Library 2.22 Reference Manual 1/2

The CERT C Coding Standard, Second Edition enumerates the coding errors that are the root causes of current software vulnerabilities in C, prioritizing them by severity, likelihood of exploitation, and remediation costs. "Secure programming in C can be more difficult than even many experienced programmers realize," said Robert C. Seacord, technical manager of the CERT Secure Coding Initiative and author of the CERT C Coding Standard. "Software systems are becoming increasing complex as our dependency on these systems increases. In our new CERT standard, as with all of our standards, we identify insecure coding practices and present secure alternatives that software developers can implement to reduce or eliminate vulnerabilities before deployment."

Python Standard Library

Using the C++ Standard Template Libraries is a contemporary treatment that teaches the generic programming capabilities that the C++ 14 Standard Library provides. In this book, author Ivor Horton explains what the class and function templates available with C++ 14 do, and how to use them in a practical context. You'll learn how to create containers, and how iterators are used with them to access, modify, and extend the data elements they contain. You'll also learn about stream iterators that can transfer data between containers and streams, including file streams. The function templates that define algorithms are explained in detail, and you'll learn how to pass function objects or lambda expressions to them to customize their behavior. Many working examples are included to demonstrate how to apply the algorithms with different types of containers. After reading this book, you will understand the scope and power of the templates that the C++ 14 Standard Library includes and how these can greatly reduce the coding and development time for many applications. You'll be able to combine the class and function templates to great effect in dealing with real-world problems. The templates in the Standard Library provide you as a C++ programmer with a comprehensive set of efficiently implemented generic programming tools that you can use for most types of application. How to use Standard Library templates with your C++ applications. Understand the different types of containers that are available and what they are used for. How to define your own class types to meet the requirements of use with containers. What iterators are, the characteristics of the various types of iterators, and how they allow algorithms to be applied to the data in different types of container. How you can define your own iterator types. What the templates that define algorithms do, and how you apply them to data stored in containers and arrays. How to access hardware clocks and use them for timing execution. How to use the templates available for compute-intensive numerical data processing. How to create and use pseudo-random number generators with distribution objects.