

Programming In C Developers Library

The Definitive Guide to Using, Programming, and Administering MySQL 5.0 and 5.1 MySQL is an open source relational database management system that has experienced a phenomenal growth in popularity and use. Known for its speed and ease of use, MySQL has proven itself to be particularly well-suited for developing database-backed websites and applications. In MySQL, Paul DuBois provides a comprehensive guide to using and administering MySQL effectively and productively. He describes everything from the basics of getting information into a database and formulating queries, to using MySQL with PHP or Perl to generate dynamic web pages, to writing your own programs that access MySQL databases, to administering MySQL servers. The fourth edition of this bestselling book has been meticulously revised and updated to thoroughly cover the latest features and capabilities of MySQL 5.0, as well as to add new coverage of features introduced with MySQL 5.1. "One of the best technical books I have read on any subject." –Gregory Haley, C Vu, The Association of C & C++ Users "A top-notch user's guide and reference manual, and in my opinion, the only book you'll need for the daily operation and maintenance of MySQL databases." –Eugene Kim, Web Techniques Introduction 1 Part I: General MySQL Use Chapter 1: Getting Started with MySQL 13 Chapter 2: Using SQL to Manage Data 101 Chapter 3: Data Types 201 Chapter 4: Stored Programs 289 Chapter 5: Query Optimization 303 Part II: Using MySQL Programming Interfaces Chapter 6: Introduction to MySQL Programming 341 Chapter 7: Writing MySQL Programs Using C 359 Chapter 8: Writing MySQL Programs Using Perl DBI 435 Chapter 9: Writing MySQL Programs Using PHP 527 Part III: MySQL Administration Chapter 10: Introduction to MySQL Administration 579 Chapter 11: The MySQL Data Directory 585 Chapter 12: General MySQL Administration 609 Chapter 13: Access Control and Security 699 Chapter 14: Database Maintenance, Backups, and Replication 737 Part IV: Appendixes Appendix A: Obtaining and Installing Software 777 Appendix B: Data Type Reference 797 Appendix C: Operator and Function Reference 813 Appendix D: System, Status, and User Variable Reference 889 Appendix E: SQL Syntax Reference 937 Appendix F: MySQL Program Reference 1037 Note: Appendixes G, H, and I are located online and are accessible either by registering this book at informit.com/register or by visiting www.kitebird.com/mysql-book. Appendix G: C API Reference 1121 Appendix H: Perl DBI API Reference 1177 Appendix I: PHP API Reference 1207 Index 1225

Beginning and experienced programmers will use this comprehensive guide to persistent memory programming. You will understand how persistent memory brings together several new software/hardware requirements, and offers great promise for better performance and faster application startup times—a huge leap forward in byte-addressable capacity compared with current DRAM offerings. This revolutionary new technology gives applications significant performance and capacity improvements over existing technologies. It requires a new way of thinking and developing, which makes this highly disruptive to the IT/computing industry. The full spectrum of industry sectors that will benefit from this technology include, but are not limited to, in-memory and traditional databases, AI, analytics, HPC, virtualization, and big data. Programming Persistent Memory describes the technology and why it is exciting the industry. It covers the operating system and hardware requirements as well as how to create development environments using emulated or real persistent memory hardware. The book explains fundamental concepts; provides an introduction to persistent memory programming APIs for C, C++, JavaScript, and other languages; discusses RMDA with persistent memory; reviews security features; and presents many examples. Source code and examples that you can run on your own systems are included. What You'll Learn Understand what persistent memory is, what it does, and the value it brings to the industry Become familiar with the operating system and hardware requirements to use persistent memory Know the fundamentals of persistent memory programming: why it is different from current programming methods, and what developers need to keep in mind when programming for persistence Look at persistent memory application development by example using the Persistent Memory Development Kit (PMDK) Design and optimize data structures for persistent memory Study how real-world applications are modified to leverage persistent memory Utilize the tools available for persistent memory programming, application performance profiling, and debugging Who This Book Is For C, C++, Java, and Python developers, but will also be useful to software, cloud, and hardware architects across a broad spectrum of sectors, including cloud service providers, independent software vendors, high performance compute, artificial intelligence, data analytics, big data, etc.

Authored by two of the leading authorities in the field, this guide offers readers the knowledge and skills needed to achieve proficiency with embedded software.

Provides instructions for writing C code to create games and mobile applications using the new C11 standard.

This is the eBook version of the print title, Framework Design Guidelines, Second Edition . Access to all the samples, applications, and content on the DVD is available through the product catalog page www.informit.com/title/9780321545619 Navigate to the "Downloads" tab and click on the "DVD Contents" links - see instructions in back pages of your eBook. Framework Design Guidelines, Second Edition, teaches developers the best practices for designing reusable libraries for the Microsoft .NET Framework. Expanded and updated for .NET 3.5, this new edition focuses on the design issues that directly affect the programmability of a class library, specifically its publicly accessible APIs. This book can improve the work of any .NET developer producing code that other developers will use. It includes copious annotations to the guidelines by thirty-five prominent architects and practitioners of the .NET Framework, providing a lively discussion of the reasons for the guidelines as well as examples of when to break those guidelines. Microsoft architects Krzysztof Cwalina and Brad Abrams teach framework design from the top down. From their significant combined experience and deep insight, you will learn The general philosophy and fundamental principles of framework design Naming guidelines for the various parts of a framework Guidelines for the design and extending of types and members of types Issues affecting—and guidelines for ensuring—extensibility How (and how not) to design exceptions Guidelines for—and examples of—common framework design patterns Guidelines in this book are presented in four major forms: Do, Consider, Avoid, and Do not. These directives help focus attention on practices that should always be used, those that should generally be used, those that should rarely be used, and those that should never be used. Every guideline includes a discussion of its applicability, and most include a code example to help illuminate the dialogue. Framework Design Guidelines, Second Edition, is the only definitive source of best practices for managed code API development, direct from the architects themselves. A companion DVD includes the Designing .NET Class Libraries video series, instructional presentations by the authors on design guidelines for developing classes and components that extend the .NET Framework. A sample API specification and other useful resources and tools are also included.

Introduces the features of the C programming language, discusses data types, variables, operators, control flow, functions, pointers, arrays, and structures, and looks at the UNIX system interface

"The security of information systems has not improved at a rate consistent with the growth and sophistication of the attacks being made against them. To address this problem, we must improve the underlying strategies and techniques used to create our systems. Specifically, we must build security in from the start, rather than append it as an afterthought. That's the point of Secure Coding in C and C++. In careful detail, this book shows software developers how to build high-quality systems that are less vulnerable to costly and even catastrophic attack. It's a book that every developer should read before the start of any serious project." --Frank Abagnale, author, lecturer, and leading consultant on fraud prevention and secure documents Learn the Root Causes of Software Vulnerabilities and How to Avoid Them Commonly exploited software vulnerabilities are usually caused by avoidable software defects. Having analyzed nearly 18,000 vulnerability reports over the past ten years, the CERT/Coordination Center (CERT/CC) has determined that a relatively small number of root causes account for most of them. This book identifies and explains these causes and shows the steps that can be taken to prevent exploitation. Moreover, this book

encourages programmers to adopt security best practices and develop a security mindset that can help protect software from tomorrow's attacks, not just today's. Drawing on the CERT/CC's reports and conclusions, Robert Seacord systematically identifies the program errors most likely to lead to security breaches, shows how they can be exploited, reviews the potential consequences, and presents secure alternatives. Coverage includes technical detail on how to improve the overall security of any C/C++ application Thwart buffer overflows and stack-smashing attacks that exploit insecure string manipulation logic Avoid vulnerabilities and security flaws resulting from the incorrect use of dynamic memory management functions Eliminate integer-related problems: integer overflows, sign errors, and truncation errors Correctly use formatted output functions without introducing format-string vulnerabilities Avoid I/O vulnerabilities, including race conditions Secure Coding in C and C++ presents hundreds of examples of secure code, insecure code, and exploits, implemented for Windows and Linux. If you're responsible for creating secure C or C++ software--or for keeping it safe--no other book offers you this much detailed, expert assistance.

Like it or not, JavaScript is everywhere these days—from browser to server to mobile—and now you, too, need to learn the language or dive deeper than you have. This concise book guides you into and through JavaScript, written by a veteran programmer who once found himself in the same position. Speaking JavaScript helps you approach the language with four standalone sections. First, a quick-start guide teaches you just enough of the language to help you be productive right away. More experienced JavaScript programmers will find a complete and easy-to-read reference that covers each language feature in depth. Complete contents include: JavaScript quick start: Familiar with object-oriented programming? This part helps you learn JavaScript quickly and properly. JavaScript in depth: Learn details of ECMAScript 5, from syntax, variables, functions, and object-oriented programming to regular expressions and JSON with lots of examples. Pick a topic and jump in. Background: Understand JavaScript's history and its relationship with other programming languages. Tips, tools, and libraries: Survey existing style guides, best practices, advanced techniques, module systems, package managers, build tools, and learning resources. Presents an introduction to Objective-C, covering such topics as classes and objects, data types, program looping, inheritance, polymorphism, variables, memory management, and archiving.

Linux for Developers shows you how to start writing great code for Linux, whether you're a Linux user with little or no coding experience, or an experienced Windows programmer. Leading IT trainer/author William "Bo" Rothwell begins with a clear and up-to-date review of modern open source software, including the licensing arrangements and tradeoffs all developers need to understand. He presents essential skills for both Linux command line and GUI environments, introducing text editors and other tools for efficient coding. Building on this knowledge, Rothwell introduces scripting tools such as Bash, Python, and Perl, as well as traditional object-oriented programming languages such as Java, C++, and C. Finally, he presents a full section on the powerful Git version control system, teaching skills you can use in Linux and many other environments. Access Linux systems, use GUIs, and work at the command line Learn how Linux organizes files and navigate its filesystem Use basic developer commands such as gzip and grep Edit programs with vi and vim, and explore alternative editors Perform basic sysadmin tasks that developers often need to handle Compare Linux languages to choose the best one for each task Write Bash scripts that interact with users or other shell features Program with Python and Perl: flow control, variables, and more Understand Linux features related to building C, C++, and Java programs Stay on top of complex projects with GIT revision control Work in GIT: staging, committing, branches, diffs, merges, and patches Manage local and remote GIT repositories This guide's modular coverage helps you quickly access whatever information you need right now.

A valuable programming reference provides a complete introduction to the Go programming language, covering all of Go's clean and easy to understand syntax and its built-in arrays, maps, slices and Unicode strings. Original.

You Will Learn C! Zed Shaw has crafted the perfect course for the beginning C programmer eager to advance their skills in any language. Follow it and you will learn the many skills early and junior programmers need to succeed—just like the hundreds of thousands of programmers Zed has taught to date! You bring discipline, commitment, persistence, and experience with any programming language; the author supplies everything else. In Learn C the Hard Way , you'll learn C by working through 52 brilliantly crafted exercises. Watch Zed Shaw's teaching video and read the exercise. Type his code precisely. (No copying and pasting!) Fix your mistakes. Watch the programs run. As you do, you'll learn what good, modern C programs look like; how to think more effectively about code; and how to find and fix mistakes far more efficiently. Most importantly, you'll master rigorous defensive programming techniques, so you can use any language to create software that protects itself from malicious activity and defects. Through practical projects you'll apply what you learn to build confidence in your new skills. Shaw teaches the key skills you need to start writing excellent C software, including Setting up a C environment Basic syntax and idioms Compilation, make files, and linkers Operators, variables, and data types Program control Arrays and strings Functions, pointers, and structs Memory allocation I/O and files Libraries Data structures, including linked lists, sort, and search Stacks and queues Debugging, defensive coding, and automated testing Fixing stack overflows, illegal memory access, and more Breaking and hacking your own C code It'll Be Hard at First. But Soon, You'll Just Get It—And That Will Feel Great! This tutorial will reward you for every minute you put into it. Soon, you'll know one of the world's most powerful programming languages. You'll be a C programmer. This guide was written for readers interested in learning the C++ programming language from scratch, and for both novice and advanced C++ programmers wishing to enhance their knowledge of C++. The text is organized to guide the reader from elementary language concepts to professional software development, with in depth coverage of all the C++ language elements en route.

Programming in Swift is a concise, carefully written tutorial on the basics of the Swift language and its use in developing iOS and OS X applications. The book makes no assumptions about prior experience with programming languages, or with Swift's precursor, Objective-C. Because of this, both beginners and experienced programmers alike can use this book to quickly and effectively learn the fundamentals of Swift programming. Readers can also learn the concepts of contemporary object-oriented programming without having to first learn all of the intricacies of a procedural language like C. This approach, combined with many small program examples and exercises at the end of each chapter, makes it ideally suited for either classroom use or self-study.

Software -- Programming Languages.

Programming with C++Programming in Objective-C 2.0Pearson Education

Authored by Roberto Ierusalimsky, 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)

Learning a language--any language--involves a process wherein you learn to rely less and less on instruction and more increasingly on the aspects of the language you've mastered. Whether you're learning French, Java, or C, at some point you'll set aside the tutorial and attempt to converse on your own. It's not necessary to know every subtle facet of French in order to speak it well, especially if there's a good dictionary available. Likewise, C programmers don't need to memorize every detail of C in order to write good programs. What they need instead is a reliable, comprehensive reference that they can keep nearby. C in a Nutshell is that reference. This long-awaited book is a complete reference to the C programming language and C runtime library. Its purpose is to serve as a convenient, reliable companion in your day-to-day work as a C programmer. C in a Nutshell covers virtually everything you need to program in C, describing all the elements of the language and illustrating their use with numerous examples. The book is divided into three distinct parts. The first part is a fast-paced description, reminiscent of the classic Kernighan & Ritchie text on which many C programmers cut their teeth. It focuses specifically on the C language and preprocessor directives, including extensions introduced to the ANSI standard in 1999. These topics and others are covered: Numeric constants Implicit and explicit type conversions Expressions and operators Functions Fixed-length and variable-length arrays Pointers Dynamic memory management Input and output The second part of the book is a comprehensive reference to the C runtime library; it includes an overview of the contents of the standard headers and a description of each standard library function. Part III provides the necessary knowledge of the C programmer's basic tools: the compiler, the make utility, and the debugger. The tools described here are those in the GNU software collection. C in a Nutshell is the perfect companion to K&R, and destined to be the most reached-for reference on your desk.

UNIX, UNIX LINUX & UNIX TCL/TK. Write software that makes the most effective use of the Linux system, including the kernel and core system libraries. The majority of both Unix and Linux code is still written at the system level, and this book helps you focus on everything above the kernel, where applications such as Apache, bash, cp, vim, Emacs, gcc, gdb, glibc, ls, mv, and X exist. Written primarily for engineers looking to program at the low level, this updated edition of Linux System Programming gives you an understanding of core internals that makes for better code, no matter where it appears in the stack. -- Provided by publisher.

Learn the C programming language from one of the best. Stephen Kochan's Programming in C is thorough with easy-to-follow instructions that are sure to benefit beginning programmers. This book provides readers with practical examples of how the C programming language can be used with small, fast programs, similar to the programming used by large game developers such as Nintendo. If you want a one-stop-source for C programming, this book is it. The book is appropriate for all introductory-to-intermediate courses on programming in the C language, including courses covering C programming for games and small-device platforms. Programming in C, Third Edition is a thoroughly revised and updated edition of Steven Kochan's classic C programming tutorial: a book that has helped thousands of students master C over the past twenty years. This edition fully reflects the latest C standard and contains current source code. It has been crafted to help students master C regardless of the platform they intend to use or the applications they intend to create -- including small-device and gaming applications, where C's elegance and speed make it especially valuable. Kochan begins with the fundamentals, then covers every facet of C language programming: variables, data types, arithmetic expressions, program looping, making decisions, arrays, functions, structures, character strings, pointers, operations on bits, the preprocessors, I/O, and more. Coverage also includes chapters on working with larger programs; debugging programs; and the fundamentals of object-oriented programming. Appendices include a complete language summary, an introduction to the Standard C Library, coverage of compiling and running programs using gcc, common programming mistakes, and more.

The free book "Fundamentals of Computer Programming with C#" is a comprehensive computer programming tutorial that teaches programming, logical thinking, data structures and algorithms, problem solving and high quality code with lots of examples in C#. It starts with the first steps in programming and software development like variables, data types, conditional statements, loops and arrays and continues with other basic topics like methods, numeral systems, strings and string processing, exceptions, classes and objects. After the basics this fundamental programming book enters into more advanced programming topics like recursion, data structures (lists, trees, hash-tables and graphs), high-quality code, unit testing and refactoring, object-oriented principles (inheritance, abstraction, encapsulation and polymorphism) and their implementation the C# language. It also covers fundamental topics that each good developer should know like algorithm design, complexity of algorithms and problem solving. The book uses C# language and Visual Studio to illustrate the programming concepts and explains some C# / .NET specific technologies like lambda expressions, extension methods and LINQ. The book is written by a team of developers lead by Svetlin Nakov who has 20+ years practical software development experience. It teaches the major programming concepts and way of thinking needed to become a good software engineer and the C# language in the meantime. It is a great start for anyone who wants to become a skillful software engineer. The books does not teach technologies like databases, mobile and web development, but shows the true way to master the basics of programming regardless of the languages, technologies and tools. It is good for beginners and intermediate developers who want to put a solid base for a successful career in the software engineering industry. The book is accompanied by free video lessons, presentation slides and mind maps, as well as hundreds of exercises and live examples. Download the free C# programming book, videos, presentations and other resources from <http://introprogramming.info>. Title: Fundamentals of Computer Programming with C# (The Bulgarian C# Programming Book) ISBN: 9789544007737 ISBN-13: 978-954-400-773-7 (9789544007737) ISBN-10: 954-400-773-3 (9544007733) Author: Svetlin Nakov & Co. Pages: 1132 Language: English Published: Sofia, 2013 Publisher: Faber Publishing, Bulgaria Web site: <http://www.introprogramming.info> License: CC-Attribution-Share-Alike Tags: free, programming, book, computer programming, programming fundamentals, ebook, book programming, C#, CSharp, C# book, tutorial, C# tutorial; programming concepts, programming fundamentals, compiler, Visual Studio, .NET, .NET Framework, data types, variables, expressions, statements, console, conditional statements, control-flow logic, loops, arrays, numeral systems, methods, strings, text processing, StringBuilder, exceptions, exception handling, stack trace, streams, files, text files, linear data structures, list, linked list, stack, queue, tree, balanced tree, graph, depth-first search, DFS, breadth-first search, BFS, dictionaries, hash tables, associative arrays, sets, algorithms, sorting algorithm, searching algorithms, recursion, combinatorial algorithms, algorithm complexity, OOP, object-oriented programming, classes, objects, constructors, fields, properties, static members, abstraction, interfaces, encapsulation, inheritance, virtual methods, polymorphism, cohesion, coupling, enumerations, generics, namespaces, UML, design patterns, extension methods, anonymous types, lambda expressions, LINQ, code quality, high-quality code, high-quality classes, high-quality methods, code formatting, self-documenting code, code refactoring, problem solving, problem solving methodology, 9789544007737, 9544007733

THE #1 BESTSELLING BOOK ON OBJECTIVE-C 2.0 Programming in Objective-C 2.0 provides the new programmer a complete, step-by-step introduction to Objective-C, the primary language used to develop applications for the iPhone, iPad, and Mac OS X platforms. The book does not assume previous experience with either C or object-oriented programming languages, and it includes many detailed, practical examples of how to put Objective-C to use in your everyday iPhone/iPad or Mac OS X programming tasks. A powerful yet simple object-oriented programming language that's based on the C programming language, Objective-C is widely available not only on OS X and the

iPhone/iPad platform but across many operating systems that support the gcc compiler, including Linux, Unix, and Windows systems. The second edition of this book thoroughly covers the latest version of the language, Objective-C 2.0. And it shows not only how to take advantage of the Foundation framework's rich built-in library of classes but also how to use the iPhone SDK to develop programs designed for the iPhone/iPad platform. Table of Contents 1 Introduction Part I: The Objective-C 2.0 Language 2 Programming in Objective-C 3 Classes, Objects, and Methods 4 Data Types and Expressions 5 Program Looping 6 Making Decisions 7 More on Classes 8 Inheritance 9 Polymorphism, Dynamic Typing, and Dynamic Binding 10 More on Variables and Data Types 11 Categories and Protocols 12 The Preprocessor 13 Underlying C Language Features Part II: The Foundation Framework 14 Introduction to the Foundation Framework 15 Numbers, Strings, and Collections 16 Working with Files 17 Memory Management 18 Copying Objects 19 Archiving Part III: Cocoa and the iPhone SDK 20 Introduction to Cocoa 21 Writing iPhone Applications Part IV: Appendixes A Glossary B Objective-C 2.0 Language Summary C Address Book Source Code D Resources

A guide to Go describes how the programming language is structured and provides examples of code that demonstrate every stage of Go development, from creating a simple program to debugging and distributing code.

Winner of the 2014 Jolt Award for "Best Book" "Whether you are an experienced programmer or are starting your career, Python in Practice is full of valuable advice and example to help you improve your craft by thinking about problems from different perspectives, introducing tools, and detailing techniques to create more effective solutions." --Doug Hellmann, Senior Developer, DreamHost If you're an experienced Python programmer, Python in Practice will help you improve the quality, reliability, speed, maintainability, and usability of all your Python programs. Mark Summerfield focuses on four key themes: design patterns for coding elegance, faster processing through concurrency and compiled Python (Cython), high-level networking, and graphics. He identifies well-proven design patterns that are useful in Python, illuminates them with expert-quality code, and explains why some object-oriented design patterns are irrelevant to Python. He also explodes several counterproductive myths about Python programming--showing, for example, how Python can take full advantage of multicore hardware. All examples, including three complete case studies, have been tested with Python 3.3 (and, where possible, Python 3.2 and 3.1) and crafted to maintain compatibility with future Python 3.x versions. All code has been tested on Linux, and most code has also been tested on OS X and Windows. All code may be downloaded at www.qtrac.eu/pipbook.html. Coverage includes Leveraging Python's most effective creational, structural, and behavioral design patterns Supporting concurrency with Python's multiprocessing, threading, and concurrent.futures modules Avoiding concurrency problems using thread-safe queues and futures rather than fragile locks Simplifying networking with high-level modules, including xmlrpclib and RPyC Accelerating Python code with Cython, C-based Python modules, profiling, and other techniques Creating modern-looking GUI applications with Tkinter Leveraging today's powerful graphics hardware via the OpenGL API using pyglet and PyOpenGL

Get started with writing simple programs in C while learning the skills that will help you work with practically any programming language Key Features Learn essential C concepts such as variables, data structures, functions, loops, and pointers Get to grips with the core programming aspects that form the base of many modern programming languages Explore the expressiveness and versatility of the C language with the help of sample programs Book Description C is a powerful general-purpose programming language that is excellent for beginners to learn. This book will introduce you to computer programming and software development using C. If you're an experienced developer, this book will help you to become familiar with the C programming language. This C programming book takes you through basic programming concepts and shows you how to implement them in C. Throughout the book, you'll create and run programs that make use of one or more C concepts, such as program structure with functions, data types, and conditional statements. You'll also see how to use looping and iteration, arrays, pointers, and strings. As you make progress, you'll cover code documentation, testing and validation methods, basic input/output, and how to write complete programs in C. By the end of the book, you'll have developed basic programming skills in C, that you can apply to other programming languages and will develop a solid foundation for you to advance as a programmer. What you will learn Understand fundamental programming concepts and implement them in C Write working programs with an emphasis on code indentation and readability Break existing programs intentionally and learn how to debug code Adopt good coding practices and develop a clean coding style Explore general programming concepts that are applicable to more advanced projects Discover how you can use building blocks to make more complex and interesting programs Use C Standard Library functions and understand why doing this is desirable Who this book is for This book is written for two very diverse audiences. If you're an absolute beginner who only has basic familiarity with operating a computer, this book will help you learn the most fundamental concepts and practices you need to know to become a successful C programmer. If you're an experienced programmer, you'll find the full range of C syntax as well as common C idioms. You can skim through the explanations and focus primarily on the source code provided.

Introduces the C programming language, covering such topics as language fundamentals, variables, data types, arithmetic expressions, program looping, functions, and arrays, with complete C programs to illustrate each new concept discussed.

C++ Primer Plus, Sixth Edition New C++11 Coverage C++ Primer Plus is a carefully crafted, complete tutorial on one of the most significant and widely used programming languages today. An accessible and easy-to-use self-study guide, this book is appropriate for both serious students of programming as well as developers already proficient in other languages. The sixth edition of C++ Primer Plus has been updated and expanded to cover the latest developments in C++, including a detailed look at the new C++11 standard. Author and educator Stephen Prata has created an introduction to C++ that is instructive, clear, and insightful. Fundamental programming concepts are explained along with details of the C++ language. Many short, practical examples illustrate just one or two concepts at a time, encouraging readers to master new topics by immediately putting them to use. Review questions and programming exercises at the end of each chapter help readers zero in on the most critical information and digest the most difficult concepts. In C++ Primer Plus, you'll find depth, breadth, and a variety of teaching techniques and tools to enhance your learning: A new detailed chapter on the changes and additional capabilities introduced in the C++11 standard Complete, integrated discussion of both basic C language and additional C++ features Clear guidance about when and why to use a feature Hands-on learning with concise and simple examples that develop your understanding a concept or two at a time Hundreds of practical sample programs Review questions and programming exercises at the end of each chapter to test your understanding Coverage

of generic C++ gives you the greatest possible flexibility Teaches the ISO standard, including discussions of templates, the Standard Template Library, the string class, exceptions, RTTI, and namespaces Table of Contents 1: Getting Started with C++ 2: Setting Out to C++ 3: Dealing with Data 4: Compound Types 5: Loops and Relational Expressions 6: Branching Statements and Logical Operators 7: Functions: C++'s Programming Modules 8: Adventures in Functions 9: Memory Models and Namespaces 10: Objects and Classes 11: Working with Classes 12: Classes and Dynamic Memory Allocation 13: Class Inheritance 14: Reusing Code in C++ 15: Friends, Exceptions, and More 16: The string Class and the Standard Template Library 17: Input, Output, and Files 18: The New C++11 Standard A Number Bases B C++ Reserved Words C The ASCII Character Set D Operator Precedence E Other Operators F The stringTemplate Class G The Standard Template Library Methods and Functions H Selected Readings and Internet Resources I Converting to ISO Standard C++ J Answers to Chapter Reviews

The Best-Selling C++ Resource Now Updated for C++11 The C++ standard library provides a set of common classes and interfaces that greatly extend the core C++ language. The library, however, is not self-explanatory. To make full use of its components—and to benefit from their power—you need a resource that does far more than list the classes and their functions. The C++ Standard Library: A Tutorial and Reference, Second Edition, describes this library as now incorporated into the new ANSI/ISO C++ language standard (C++11). The book provides comprehensive documentation of each library component, including an introduction to its purpose and design; clearly written explanations of complex concepts; the practical programming details needed for effective use; traps and pitfalls; the exact signature and definition of the most important classes and functions; and numerous examples of working code. The book focuses in particular on the Standard Template Library (STL), examining containers, iterators, function objects, and STL algorithms. The book covers all the new C++11 library components, including Concurrency Fractional arithmetic Clocks and timers Tuples New STL containers New STL algorithms New smart pointers New locale facets Random numbers and distributions Type traits and utilities Regular expressions The book also examines the new C++ programming style and its effect on the standard library, including lambdas, range-based for loops, move semantics, and variadic templates. An accompanying Web site, including source code, can be found at www.cppstdlib.com.

The perfect beginner's guide to Objective-C 2.0, the essential language for over 1,000,000 Mac OS X, iPhone, and iPod touch developers! • Concise, readable, and friendly: designed to get new Objective-C programmers up and running fast! • Covers everything readers need to know, from basic Object-Oriented Programming to general C concepts. • Walks through code examples one line at a time, and also offers high-level explanations what's happening 'behind the scenes' of Objective-C programs. Long-time OS X and iPhone developer Robert Clair begins with a concise review of the object-oriented and C concepts that all Objective-C developers need to know. Next, he introduces the basics of the Objective-C language, walking through code examples one line at a time, and offering high-level explanations of what's happening 'behind the scenes.' Clair concludes with advanced topics carefully chosen for their real-world value - including detailed coverage of memory management and the differences between 32-bit and 64-bit programs. Throughout, Learning Objective-C 2.0 focuses consistently on the features, concepts, and techniques that matter most in day-to-day programming - not complex 'edge cases' or abstract theory. The result: an outstanding first book for every beginner who wants to program for Apple's fast-growing iPhone and Mac OS X platforms. Note: This will be the entry-level book for Objective-C newcomers. Readers who complete it can move on to Stephen Kochan's highly-regarded Programming in Objective-C 2.0 and then to our more specialized Apple development titles, such as David Chisnall's Cocoa Developer's Handbook, Fritz Anderson Xcode 3.x Unleashed , and Aaron Hillegass's Cocoa Programming for Mac OS X Third Ed Tap into the wisdom of experts to learn what every programmer should know, no matter what language you use. With the 97 short and extremely useful tips for programmers in this book, you'll expand your skills by adopting new approaches to old problems, learning appropriate best practices, and honing your craft through sound advice. With contributions from some of the most experienced and respected practitioners in the industry--including Michael Feathers, Pete Goodliffe, Diomidis Spinellis, Cay Horstmann, Verity Stob, and many more--this book contains practical knowledge and principles that you can apply to all kinds of projects. A few of the 97 things you should know: "Code in the Language of the Domain" by Dan North "Write Tests for People" by Gerard Meszaros "Convenience Is Not an -ility" by Gregor Hohpe "Know Your IDE" by Heinz Kabutz "A Message to the Future" by Linda Rising "The Boy Scout Rule" by Robert C. Martin (Uncle Bob) "Beware the Share" by Udi Dahan

Sams Teach Yourself C Programming in One Hour a Day, Seventh Edition is the newest version of the worldwide best-seller Sams Teach Yourself C in 21 Days. Fully revised for the new C11 standard and libraries, it now emphasizes platform-independent C programming using free, open-source C compilers. This edition strengthens its focus on C programming fundamentals, and adds new material on popular C-based object-oriented programming languages such as Objective-C. Filled with carefully explained code, clear syntax examples, and well-crafted exercises, this is the broadest and deepest introductory C tutorial available. It's ideal for anyone who's serious about truly mastering C – including thousands of developers who want to leverage its speed and performance in modern mobile and gaming apps. Friendly and accessible, it delivers step-by-step, hands-on experience that starts with simple tasks and gradually builds to professional-quality techniques. Each lesson is designed to be completed in hour or less, introducing and clearly explaining essential concepts, providing practical examples, and encouraging you to build simple programs on your own. Coverage includes: Understanding C program components and structure Mastering essential C syntax and program control Using core language features, including numeric arrays, pointers, characters, strings, structures, and variable scope Interacting with the screen, printer, and keyboard Using functions and exploring the C Function Library Working with memory and the compiler Contents at a Glance PART I: FUNDAMENTALS OF C 1 Getting Started with C 2 The Components of a C Program 3 Storing Information: Variables and Constants 4 The Pieces of a C Program: Statements,

Expressions, and Operators 5 Packaging Code in Functions 6 Basic Program Control 7 Fundamentals of Reading and Writing Information PART II: PUTTING C TO WORK 8 Using Numeric Arrays 9 Understanding Pointers 10 Working with Characters and Strings 11 Implementing Structures, Unions, and TypeDefs 12 Understanding Variable Scope 13 Advanced Program Control 14 Working with the Screen, Printer, and Keyboard PART III: ADVANCED C 15 Pointers to Pointers and Arrays of Pointers 16 Pointers to Functions and Linked Lists 17 Using Disk Files 18 Manipulating Strings 19 Getting More from Functions 20 Exploring the C Function Library 21 Working with Memory 22 Advanced Compiler Use PART IV: APPENDIXES A ASCII Chart B C/C++ Reserved Words C Common C Functions D Answers

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.

Cocoa Programming is a comprehensive work that starts as a fast-paced introduction to the OS architecture and the Cocoa language for those programmers new to the environment. The more advanced sections of the book will show the reader how to create Cocoa applications using Objective-C, to modify the views, integrate multimedia, and access networks. The final sections of the book explain how to extend system applications and development tools in order to create your own frameworks.

If you are new to C++ programming, *C++ Primer Plus, Fifth Edition* is a friendly and easy-to-use self-study guide. You will cover the latest and most useful language enhancements, the Standard Template Library and ways to streamline object-oriented programming with C++. This guide also illustrates how to handle input and output, make programs perform repetitive tasks, manipulate data, hide information, use functions and build flexible, easily modifiable programs. With the help of this book, you will: Learn C++ programming from the ground up. Learn through real-world, hands-on examples. Experiment with concepts, including classes, inheritance, templates and exceptions. Reinforce knowledge gained through end-of-chapter review questions and practice programming exercises. *C++ Primer Plus, Fifth Edition* makes learning and using important object-oriented programming concepts understandable. Choose this classic to learn the fundamentals and more of C++ programming.

What others in the trenches say about *The Pragmatic Programmer*... "The cool thing about this book is that it's great for keeping the programming process fresh. The book helps you to continue to grow and clearly comes from people who have been there." —Kent Beck, author of *Extreme Programming Explained: Embrace Change* "I found this book to be a great mix of solid advice and wonderful analogies!" —Martin Fowler, author of *Refactoring* and *UML Distilled* "I would buy a copy, read it twice, then tell all my colleagues to run out and grab a copy. This is a book I would never loan because I would worry about it being lost." —Kevin Ruland, Management Science, MSG-Logistics "The wisdom and practical experience of the authors is obvious. The topics presented are relevant and useful.... By far its greatest strength for me has been the outstanding analogies—tracer bullets, broken windows, and the fabulous helicopter-based explanation of the need for orthogonality, especially in a crisis situation. I have little doubt that this book will eventually become an excellent source of useful information for journeymen programmers and expert mentors alike." —John Lakos, author of *Large-Scale C++ Software Design* "This is the sort of book I will buy a dozen copies of when it comes out so I can give it to my clients." —Eric Vought, Software Engineer "Most modern books on software development fail to cover the basics of what makes a great software developer, instead spending their time on syntax or technology where in reality the greatest leverage possible for any software team is in having talented developers who really know their craft well. An excellent book." —Pete McBreen, Independent Consultant "Since reading this book, I have implemented many of the practical suggestions and tips it contains. Across the board, they have saved my company time and money while helping me get my job done quicker! This should be a desktop reference for everyone who works with code for a living." —Jared Richardson, Senior Software Developer, iRenaissance, Inc. "I would like to see this issued to every new employee at my company...." —Chris Cleeland, Senior Software Engineer, Object Computing, Inc. "If I'm putting together a project, it's the authors of this book that I want. . . . And failing that I'd settle for people who've read their book." —Ward Cunningham

Straight from the programming trenches, *The Pragmatic Programmer* cuts through the increasing specialization and technicalities of modern software development to examine the core process--taking a requirement and producing working, maintainable code that delights its users. It covers topics ranging from personal responsibility and career development to architectural techniques for keeping your code flexible and easy to adapt and reuse. Read this book, and you'll learn how to Fight software rot; Avoid the trap of duplicating knowledge; Write flexible, dynamic, and adaptable code; Avoid programming by coincidence; Bullet-proof your code with contracts, assertions, and exceptions; Capture real requirements; Test ruthlessly and effectively; Delight your users; Build teams of pragmatic programmers; and Make your developments more precise with automation. Written as a series of self-contained sections and filled with entertaining anecdotes, thoughtful examples, and interesting analogies, *The Pragmatic Programmer* illustrates the best practices and major pitfalls of many different aspects of software development. Whether you're a new coder, an experienced programmer, or a manager responsible for software projects, use these lessons daily, and you'll quickly see improvements in personal productivity, accuracy, and job satisfaction. You'll learn skills and develop habits and attitudes that form the foundation for long-term success in your career. You'll become a Pragmatic Programmer.

Peter Seibel interviews 15 of the most interesting computer programmers alive today in *Coders at Work*, offering a companion volume to Apress's highly acclaimed best-seller *Founders at Work* by Jessica Livingston. As the words "at work" suggest, Peter

Seibel focuses on how his interviewees tackle the day-to-day work of programming, while revealing much more, like how they became great programmers, how they recognize programming talent in others, and what kinds of problems they find most interesting. Hundreds of people have suggested names of programmers to interview on the Coders at Work web site: www.codersatwork.com. The complete list was 284 names. Having digested everyone's feedback, we selected 15 folks who've been kind enough to agree to be interviewed: Frances Allen: Pioneer in optimizing compilers, first woman to win the Turing Award (2006) and first female IBM fellow Joe Armstrong: Inventor of Erlang Joshua Bloch: Author of the Java collections framework, now at Google Bernie Cosell: One of the main software guys behind the original ARPANET IMPs and a master debugger Douglas Crockford: JSON founder, JavaScript architect at Yahoo! L. Peter Deutsch: Author of Ghostscript, implementer of Smalltalk-80 at Xerox PARC and Lisp 1.5 on PDP-1 Brendan Eich: Inventor of JavaScript, CTO of the Mozilla Corporation Brad Fitzpatrick: Writer of LiveJournal, OpenID, memcached, and Perlbal Dan Ingalls: Smalltalk implementor and designer Simon Peyton Jones: Coinventor of Haskell and lead designer of Glasgow Haskell Compiler Donald Knuth: Author of The Art of Computer Programming and creator of TeX Peter Norvig: Director of Research at Google and author of the standard text on AI Guy Steele: Coinventor of Scheme and part of the Common Lisp Gang of Five, currently working on Fortress Ken Thompson: Inventor of UNIX Jamie Zawinski: Author of XEmacs and early Netscape/Mozilla hacker

Offers more than one hundred customizable code phrases for Objective-C programming projects.

[Copyright: 20bad5ba294ee790f01d045ada1a793e](#)