The Cert C Coding Standard Second Edition 98 Rules For Developing Safe Reliable And Secure Systems Sei Series In Software Engineering

Password sniffing, spoofing, buffer overflows, and denial of service: these are only a few of the attacks on today's computer systems and networks. At the root of this epidemic is poorly written, poorly tested, and insecure code that puts everyone at risk. Clearly, today's developers need help figuring out how to write code that attackers won't be able to exploit. But writing such code is surprisingly difficult. Secure Programming Cookbook for C and C++ is an important new resource for developers serious about writing secure code. It contains a wealth of solutions to problems faced by those who care about the security of their applications. It covers a wide range of topics, including safe initialization, access control, input validation, symmetric and public key cryptography, cryptographic hashes and MACs, authentication and key exchange, PKI, random numbers, and anti-tampering. The rich set of code samples provided in the book's more than 200 recipes will help programmers secure the C and C++ programs they write for both Unix® (including Linux®) and Windows® environments. Readers will learn: How to avoid common programming errors, such as buffer overflows, race conditions, and format string problems How to properly SSL-enable applications How to create secure channels for client-server communication without SSL How to integrate Public Key Infrastructure (PKI) into applications Best practices for using cryptography properly Techniques and strategies for properly validating input to programs How to launch programs securely How to use file access mechanisms properly Techniques for protecting applications from reverse engineering The book's web site supplements the book by providing a place to post new recipes, including those written in additional languages like Perl, Java, and Python. Monthly prizes will reward the best recipes submitted by readers. Secure Programming Cookbook for C and C++ is destined to become an essential part of any developer's library, a code companion developers will turn to again and again as they seek to protect their systems from attackers and reduce the risks they face in today's dangerous world.

For courses in computer programming C How to Program is a comprehensive introduction to programming in C. Like other texts of the Deitels' How to Program series, the book serves as a detailed beginner source of information for college students looking to embark on a career in coding, or instructors and software-development professionals seeking to learn how to program with C. The Eighth Edition continues the tradition of the signature Deitel "Live Code" approach--presenting concepts in the context of full-working programs rather than incomplete snips of code. This gives readers a chance to run each program as they study it and see how their learning applies to real world programming scenarios.

Test your knowledge and know what to expect on A+ exam day CompTIA A+ Complete Practice Tests, Second Edition enables you to hone your test-taking skills, focus on challenging areas, and be thoroughly prepared to ace the exam and earn your A+ certification. This essential component of your overall study plan presents nine unique practice tests—and two 90-question bonus tests—covering 100% of the objective domains for both the 220-1001 and 220-1002 exams. Comprehensive coverage of every essential exam topic ensures that you will know what to expect on exam day and maximize your chances for success. Over 1200 practice questions on topics including hardware, networking, mobile devices, operating systems and procedures, troubleshooting, and more, lets you assess your performance and gain the confidence you need to pass the exam with flying colors. This second edition has been fully updated to reflect the latest best practices and updated exam objectives you will see on the big day. A+ certification is a crucial step in your IT career. Many businesses require this

accreditation when hiring computer technicians or validating the skills of current employees. This collection of practice tests allows you to: Access the test bank in the Sybex interactive learning environment Understand the subject matter through clear and accurate answers and explanations of exam objectives Evaluate your exam knowledge and concentrate on problem areas Integrate practice tests with other Sybex review and study guides, including the CompTIA A+ Complete Study Guide and the CompTIA A+ Complete Deluxe Study Guide Practice tests are an effective way to increase comprehension, strengthen retention, and measure overall knowledge. The CompTIA A+ Complete Practice Tests, Second Edition is an indispensable part of any study plan for A+ certification.

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 tens of thousands of vulnerability reports since 1988, CERT has determined that a relatively small number of root causes account for most of the vulnerabilities. Secure Coding in C and C++, Second Edition, identifies and explains these root causes and shows the steps that can be taken to prevent exploitation. Moreover, this book encourages programmers to adopt security best practices and to develop a security mindset that can help protect software from tomorrow's attacks, not just today's. Drawing on the CERT's reports and conclusions, Robert C. 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 or C++ application Thwart buffer overflows, stack-smashing, and return-oriented programming 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 resulting from signed integer overflows, unsigned integer wrapping, and truncation errors Perform secure I/O, avoiding file system vulnerabilities Correctly use formatted output functions without introducing format-string vulnerabilities Avoid race conditions and other exploitable vulnerabilities while developing concurrent code The second edition features Updates for C11 and C++11 Significant revisions to chapters on strings, dynamic memory management, and integer security A new chapter on concurrency Access to the online secure coding course offered through Carnegie Mellon's Open Learning Initiative (OLI) Secure Coding in C and C++, Second Edition, 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.

The definitive reference for any C++ programmer or for programmers needing to work with C++ programs. Every book written about C++ refers frequently to the international standard that defines the language, this will be a must-have companion volume for everyone who is serious about programming in this language. The complete C++ standard as approved by international standards bodies (BSI and ANSI) The ONLY available bound version of the standard Foreword by Bjarne Stroustrup Most recent corrections and updates (Technical Corrigendum) are indicated with side bars to highlight where changes have taken place An introductory chapter explains what the standards process is and how the reader can participate in the standards process

"At Cisco, we have adopted the CERT C Coding Standard as the internal secure coding standard for all C developers. It is a core component of our secure development lifecycle. The coding standard described in this book breaks down complex software security topics into easy-to-follow rules with excellent real-world examples. It is an essential reference for any developer who wishes to write secure and resilient software in C and C++." —Edward D. Paradise, vice president, engineering, threat response, intelligence, and development, Cisco Systems Secure programming in C can be more difficult than even many experienced programmers realize. To help programmers write more secure

code, The CERT® C Coding Standard, Second Edition, fully documents the second official release of the CERT standard for secure coding in C. The rules laid forth in this new edition will help ensure that programmers' code fully complies with the new C11 standard; it also addresses earlier versions, including C99. The new standard itemizes those coding errors that are the root causes of current software vulnerabilities in C, prioritizing them by severity, likelihood of exploitation, and remediation costs. Each of the text's 98 guidelines includes examples of insecure code as well as secure, C11-conforming, alternative implementations. If uniformly applied, these guidelines will eliminate critical coding errors that lead to buffer overflows, format-string vulnerabilities, integer overflow, and other common vulnerabilities. This book reflects numerous experts' contributions to the open development and review of the rules and recommendations that comprise this standard. Coverage includes Preprocessor Declarations and Initialization Expressions Integers Floating Point Arrays Characters and Strings Memory Management Input/Output Environment Signals Error Handling Concurrency Miscellaneous Issues

A comprehensive guide to understanding the language of C offers solutions for everyday programming tasks and provides all the necessary information to understand and use common programming techniques. Original. (Intermediate).

PGP is a freely available encryption program that protects the privacy of files and electronic mail. It uses powerful public key cryptography and works on virtually every platform. This book is both a readable technical user's guide and a fascinating behind-the-scenes look at cryptography and privacy. It describes how to use PGP and provides background on cryptography, PGP's history, battles over public key cryptography patents and U.S. government export restrictions, and public debates about privacy and free speech.

Consisting of selected papers from the third international conference on Future Generation Information Technology (FGIT 2011), this volume focuses on the various aspects of advances in information technology.

Human factors and usability issues have traditionally played a limited role in security research and secure systems development. Security experts have largely ignored usability issues--both because they often failed to recognize the importance of human factors and because they lacked the expertise to address them. But there is a growing recognition that today's security problems can be solved only by addressing issues of usability and human factors. Increasingly, well-publicized security breaches are attributed to human errors that might have been prevented through more usable software. Indeed, the world's future cyber-security depends upon the deployment of security technology that can be broadly used by untrained computer users. Still, many people believe there is an inherent tradeoff between computer security and usability. It's true that a computer without passwords is usable, but not very secure. A computer that makes you authenticate every five minutes with a password and a fresh drop of blood might be very secure, but nobody would use it. Clearly, people need computers, and if they can't use one that's secure, they'll use one that isn't. Unfortunately, unsecured systems aren't usable for long, either. They get hacked, compromised, and otherwise rendered useless. There is increasing agreement that we need to design secure systems that people can actually use, but less agreement about how to reach this goal. Security & Usability is the first book-length work describing the current state of the art in this emerging field. Edited by security experts Dr. Lorrie Faith Cranor and Dr. Simson Garfinkel, and authored by cutting-edge

security and human-computerinteraction (HCI) researchers world-wide, this volume is expected to become both a classic reference and an inspiration for future research. Security & Usability groups 34 essays into six parts: Realigning Usability and Security---with careful attention to user-centered design principles, security and usability can be synergistic. Authentication Mechanisms-- techniques for identifying and authenticating computer users. Secure Systems--how system software can deliver or destroy a secure user experience. Privacy and Anonymity Systems--methods for allowing people to control the release of personal information. Commercializing Usability: The Vendor Perspective--specific experiences of security and software vendors (e.g.,IBM, Microsoft, Lotus, Firefox, and Zone Labs) in addressing usability. The Classics--groundbreaking papers that sparked the field of security and usability. This book is expected to start an avalanche of discussion, new ideas, and further advances in this important field.

In a concise and direct question-and-answer format, C++ FAQs, Second Edition brings you the most efficient solutions to more than four hundred of the practical programming challenges you face every day. Moderators of the on-line C++ FAQ at comp.lang.c++, Marshall Cline, Greg Lomow, and Mike Girou are familiar with C++ programmers' most pressing concerns. In this book, the authors concentrate on those issues most critical to the professional programmer's work, and they present more explanatory material and examples than is possible on-line. This book focuses on the effective use of C++, helping programmers avoid combining seemingly legal C++ constructs in incompatible ways. This second edition is completely up-to-date with the final ANSI/ISO C++ Standard. It covers some of the smaller syntax changes, such as "mutable"; more significant changes, such as RTTI and namespaces; and such major innovations as the C++ Standard Library, including the STL. In addition, this book discusses technologies such as Java, CORBA, COM/COM+, and ActiveX—and the relationship all of these have with C++. These new features and technologies are iconed to help you quickly find what is new and different in this edition. Each question-and-answer section contains an overview of the problem and solution, fuller explanations of concepts, directions for proper use of language features, guidelines for best practices and practices to avoid, and plenty of working, stand-alone examples. This edition is thoroughly cross-referenced and indexed for quick access. Get a value-added service! Try out all the examples from this book at www.codesaw.com. CodeSaw is a free online learning tool that allows you to experiment with live code from your book right in your browser.

“I’m an enthusiastic supporter of the CERT Secure Coding Initiative. Programmers have lots of sources of advice on correctness, clarity, maintainability, performance, and even safety. Advice on how specific language features affect security has been missing. The CERT® C Secure Coding Standard fills this need.” –Randy Meyers, Chairman of ANSI C

“For years we have relied upon the CERT/CC to publish advisories documenting an endless stream of security problems.

Now CERT has embodied the advice of leading technical experts to give programmers and managers the practical guidance needed to avoid those problems in new applications and to help secure legacy systems. Well done!" –Dr. Thomas Plum, founder of Plum Hall, Inc. "Connectivity has sharply increased the need for secure, hacker-safe applications. By combining this CERT standard with other safety guidelines, customers gain all-round protection and approach the goal of zero-defect software." –Chris Tapp, Field Applications Engineer, LDRA Ltd. "I've found this standard to be an indispensable collection of expert information on exactly how modern software systems fail in practice. It is the perfect place to start for establishing internal secure coding guidelines. You won't find this information elsewhere, and, when it comes to software security, what you don't know is often exactly what hurts you." –John McDonald, coauthor of The Art of Software Security Assessment Software security has major implications for the operations and assets of organizations, as well as for the welfare of individuals. To create secure software, developers must know where the dangers lie. Secure programming in C can be more difficult than even many experienced programmers believe. This book is an essential desktop reference documenting the first official release of The CERT® C Secure Coding Standard . The standard itemizes those coding errors that are the root causes of software vulnerabilities in C and prioritizes them by severity, likelihood of exploitation, and remediation costs. Each guideline provides examples of insecure code as well as secure, alternative implementations. If uniformly applied, these guidelines will eliminate the critical coding errors that lead to buffer overflows, format string vulnerabilities, integer overflow, and other common software vulnerabilities.

"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.

PRACTICAL, EXAMPLE-RICH COVERAGE OF: Classes, Objects, Encapsulation, Inheritance, Polymorphism Integrated OOP Case Studies: Time, GradeBook, Employee Industrial-Strength, 95-Page OOD/UML® 2 ATM Case Study Standard Template Library (STL): Containers, Iterators and Algorithms I/O, Types, Control Statements, Functions Arrays, Vectors, Pointers, References String Class, C-Style Strings Operator Overloading, Templates Exception Handling, Files Bit and Character Manipulation Boost Libraries and the Future of C++ GNU™ and Visual C++® Debuggers And more… VISIT WWW.DEITEL.COM For information on Deitel® Dive-Into® Series corporate training courses offered at customer sites worldwide (or write to deitel@deitel.com) Download code examples Check out the growing list of programming, Web 2.0 and software-related Resource Centers To receive updates for this book, subscribe to the free DEITEL® BUZZ ONLINE e-mail newsletter at www.deitel.com/newsletter/subscribe.html Read archived issues of the DEITEL® BUZZ ONLINE The professional programmer’s DEITEL® guide to C++ and object-oriented application development Written for programmers with a background in high-level language programming, this book applies the Deitel signature live-code approach to teaching programming and explores the C++ language and C++ Standard Libraries in depth. The book presents the concepts in the context of fully tested programs, complete with syntax shading, code highlighting, code walkthroughs and program outputs. The book features 240 C++ applications with over 15,000 lines of proven C++ code, and hundreds of tips that will help you build robust applications. Start with an introduction to C++ using an early classes and objects approach, then rapidly move on to more advanced topics, including templates, exception handling, the Standard Template Library (STL) and selected features from the Boost libraries. You’ll enjoy the Deitels’ classic treatment of object-oriented programming and the OOD/UML® 2 ATM case study, including a complete C++ implementation. When you’re finished, you’ll have everything you need to build object-oriented C++ applications. The

DEITEL® Developer Series is designed for practicing programmers. The series presents focused treatments of emerging technologies, including C++, .NET, Java™, web services, Internet and web development and more. PRE-PUBLICATION REVIEWER TESTIMONIALS “An excellent ‘objects first’ coverage of C++. The example-driven presentation is enriched by the optional UML case study that contextualizes the material in an ongoing software engineering project.” –Gavin Osborne, Saskatchewan Institute of Applied Science and Technology “Introducing the UML early on is a great idea.” –Raymond Stephenson, Microsoft “Good use of diagrams, especially of the activation call stack and recursive functions.” –Amar Raheja, California State Polytechnic University, Pomona “Terrific discussion of pointers–probably the best I have seen.” –Anne B. Horton, Lockheed Martin “Great coverage of polymorphism and how the compiler implements polymorphism ‘under the hood.’” –Ed James-Beckham, Borland “The Boost/C++0x chapter will get you up and running quickly with the memory management and regular expression libraries, plus whet your appetite for new C++ features being standardized.” –Ed Brey, Kohler Co. “Excellent introduction to the Standard Template Library (STL). The best book on C++ programming!” –Richard Albright, Goldey-Beacom College “Just when you think you are focused on learning one topic, suddenly you discover you’ve learned more than you expected.” –Chad Willwerth, University of Washington, Tacoma “The most thorough C++ treatment I’ve seen. Replete with real-world case studies covering the full software development lifecycle. Code examples are extraordinary!” –Terrell Hull, Logicalis Integration Solutions/

A principal source of risk in component-based software design, say Wallnau and two other technicians at the institute, Scott A. Hissam and Robert C. Seacord, is a lack of knowledge about how components should be integrated and how they behave when integrated. To mitigate that risk, they introduce several concepts, among them the component ensemble as a design abstraction, blackboards as a fundamental design notation, and a process for exposing design risk. They speak to practicing and student software engineers. c. Book News Inc.

Barr Group's Embedded C Coding Standard was developed to help firmware engineers minimize defects in embedded systems. Unlike the majority of coding standards, this standard focuses on practical rules that keep bugs out - including techniques designed to improve the maintainability and portability of embedded software. The rules in this coding standard include a set of guiding principles, as well as specific naming conventions and other rules for the use of data types, functions, preprocessor macros, variables, and other C language constructs. Individual rules that have been demonstrated to reduce or eliminate certain types of defects are highlighted. The BARR-C standard is distinct from, yet compatible with, the MISRA C Guidelines for Use of the C Language in Critical Systems. Programmers can easily combine rules from the two standards as needed.

Cyber Security Engineering is the definitive modern reference and tutorial on the full range of capabilities associated with modern

cyber security engineering. Pioneering software assurance experts Dr. Nancy R. Mead and Dr. Carol C. Woody bring together comprehensive best practices for building software systems that exhibit superior operational security, and for considering security throughout your full system development and acquisition lifecycles. Drawing on their pioneering work at the Software Engineering Institute (SEI) and Carnegie Mellon University, Mead and Woody introduce seven core principles of software assurance, and show how to apply them coherently and systematically. Using these principles, they help you prioritize the wide range of possible security actions available to you, and justify the required investments. Cyber Security Engineering guides you through risk analysis, planning to manage secure software development, building organizational models, identifying required and missing competencies, and defining and structuring metrics. Mead and Woody address important topics, including the use of standards, engineering security requirements for acquiring COTS software, applying DevOps, analyzing malware to anticipate future vulnerabilities, and planning ongoing improvements. This book will be valuable to wide audiences of practitioners and managers with responsibility for systems, software, or quality engineering, reliability, security, acquisition, or operations. Whatever your role, it can help you reduce operational problems, eliminate excessive patching, and deliver software that is more resilient and secure.
Teach Your Students How to Program Well Intermediate C Programming provides a stepping-stone for intermediate-level students to go from writing short programs to writing real programs well. It shows students how to identify and eliminate bugs, write clean code, share code with others, and use standard Linux-based tools, such as ddd and valgrind. The text covers numerous concepts and tools that will help your students write better programs. It enhances their programming skills by explaining programming concepts and comparing common mistakes with correct programs. It also discusses how to use debuggers and the strategies for debugging as well as studies the connection between programming and discrete mathematics.
Unlike high-level languages such as Java and C++, assembly language is much closer to the machine code that actually runs computers; it's used to create programs or modules that are very fast and efficient, as well as in hacking exploits and reverse engineering Covering assembly language in the Pentium microprocessor environment, this code-intensive guide shows programmers how to create stand-alone assembly language programs as well as how to incorporate assembly language libraries or routines into existing high-level applications Demonstrates how to manipulate data, incorporate advanced functions and libraries, and maximize application performance Examples use C as a high-level language, Linux as the development environment, and GNU tools for assembling, compiling, linking, and debugging
Despite their myriad manifestations and different targets, nearly all attacks on computer systems have one fundamental cause: the code used to run far too many systems today is not secure. Flaws in its design, implementation, testing, and operations allow attackers all-too-easy access. "Secure Coding, by Mark G. Graff and Ken vanWyk, looks at the problem of bad code in a new way. Packed with advice based on the authors' decades of experience in the computer security field, this concise and highly readable book explains why so much code today is filled with vulnerabilities, and tells readers what they must do to avoid writing code that can be exploited by attackers. Beyond the technical, "Secure Coding sheds new light on the economic, psychological, and sheer

practical reasons why security vulnerabilities are so ubiquitous today. It presents a new way of thinking about these vulnerabilities and ways that developers can compensate for the factors that have produced such unsecured software in the past. It issues a challenge to all those concerned about computer security to finally make a commitment to building code the right way.
The expert guide to building Ruby on Rails applications Ruby on Rails strips complexity from the development process, enabling professional developers to focus on what matters most: delivering business value. Now, for the first time, there's a comprehensive, authoritative guide to building production-quality software with Rails. Pioneering Rails developer Obie Fernandez and a team of experts illuminate the entire Rails API, along with the Ruby idioms, design approaches, libraries, and plug-ins that make Rails so valuable. Drawing on their unsurpassed experience, they address the real challenges development teams face, showing how to use Rails' tools and best practices to maximize productivity and build polished applications users will enjoy. Using detailed code examples, Obie systematically covers Rails' key capabilities and subsystems. He presents advanced programming techniques, introduces open source libraries that facilitate easy Rails adoption, and offers important insights into testing and production deployment. Dive deep into the Rails codebase together, discovering why Rails behaves as it does— and how to make it behave the way you want it to. This book will help you Increase your productivity as a web developer Realize the overall joy of programming with Ruby on Rails Learn what's new in Rails 2.0 Drive design and protect long-term maintainability with TestUnit and RSpec Understand and manage complex program flow in Rails controllers Leverage Rails' support for designing REST-compliant APIs Master sophisticated Rails routing concepts and techniques Examine and troubleshoot Rails routing Make the most of ActiveRecord object-relational mapping Utilize Ajax within your Rails applications Incorporate logins and authentication into your application Extend Rails with the best third-party plug-ins and write your own Integrate email services into your applications with ActionMailer Choose the right Rails production configurations Streamline deployment with Capistrano
Become a pro at securing your Python apps with this step-by-step guideAbout This Book* Get the only book on the market that will help you master Python security* Make your programs more robust, secure, and safe for complex-level applications* This book provides various approaches to securing code that will enable you to implement solutions from the word goWho This Book Is ForThis book is aimed at Python developers who want to make their programs secure. Basic knowledge of Python is expected.What You Will Learn* Simulate various attack scenarios* Perform vulnerability testing using various tools and techniques* Use bruteforce to automate data mining* Identify and mitigate attacks with various OWASP projects* Perform recon and scanning automation to build your own security toolkit* Find about phishing and fuzzing in Python* Conduct network forensic analysis and packet analysis* Work through offensive programming techniques to keep your code clean and preciseIn DetailPython is used for a lot of applications, ranging from building web applications and enterprise application to the world of big data. With everyday attacks on applications by hackers, securing applications has become a critical component for Python developers.Starting with the basics to ensure the fundamentals required for security, you will gradually move on to automating various web application attacks, which can then be used by security engineers to perform automated tests. You will mitigate

various application security vulnerabilities and explore the defense mechanisms available for developers in Python. You will then learn about the various phases of network security testing that can be automated and how an engineer can simulate various attacks in controlled manner to scan for vulnerabilities. Next, you will learn how to automate password cracking using Python and focus on fuzzing, a key concept of exploit writing and protocol analysis.After reading this book, you will be able to secure your programs and applications and be ready for any kind of spyware and malware.

The CERT C Secure Coding StandardPearson Education

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 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."

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

Covers topics such as the importance of secure systems, threat modeling, canonical representation issues, solving database input, denial-of-service attacks, and security code reviews and checklists.

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.

This report describes the results of a study to evaluate the effectiveness of secure coding practices, including the use of static analysis tools coupled with secure coding rule sets such as the CERT C Programming Language Secure Coding Standard (CERT 07a) and the CERT C++ Programming Language Secure Coding Standard (CERT 07b). This study represents a joint effort between the CERT Secure Coding Initiative and JPCERT/CC. The CERT Secure Coding Initiative was established to work with software developers and software development organizations to eliminate vulnerabilities resulting from coding errors before they are deployed. The goal of this effort is to reduce the number of vulnerabilities to a level where they can be handled by existing vulnerability analysis teams around the world and decrease remediation costs by eliminating vulnerabilities before software is deployed. JPCERT/CC is the first CSIRT (computer security incident response team) established in Japan. The objectives of the study were to evaluate the efficacy of the CERT Secure Coding Standards and source code analysis tools in improving the quality and security of commercial software projects. Two static analysis tools, Fortify Source Code Analysis (SCA) from Fortify Software and Compass/ROSE from Lawrence Livermore National Laboratory were selected for their extensibility as well as overall effectiveness. Checkers were then developed for each of the tools to check code for violations of the CERT C and C++ Secure Coding Standards. The tools were then provided to Software Research Associates, Inc., Japan, which evaluated the extended versions of Fortify SCA and Compass/ROSE on two existing projects: an electronic toll collection (ETC) system-related GUI application written in C++ and an IP-TV Service Protocol Stack (IP-TV) written in the C programming language. The project successfully extended source code analysis tools to discover software defects in both projects evaluated.

An essential element of secure coding in the C programming language is well documented and enforceable coding standards. Coding standards encourage programmers to follow a uniform set of rules and guidelines determined by the requirements of the project and organization, rather than by the programmer's familiarity or preference. Once established, these standards can be used as a metric to evaluate source code (using manual or automated processes). The CERT C Secure Coding Standard provides rules and recommendations for secure coding in the C programming language. The goal of these rules and recommendations is to eliminate insecure coding practices and undefined behaviours that can lead to exploitable vulnerabilities. The application of the secure coding standard will lead to higher-quality systems that are robust and more resistant to attack. The book is intended to be used as a reference by both programming teams and individuals. it is based on the web site created for this standard. While the web site will be dynamic, organizations will need a reference that's fixed in time.

Due to the continuously stream of security breaches two security architects in the Netherlands started a project to harvest good practices for better and faster creating architecture and privacy solution designs. This project resulted in a reference architecture that is aimed to help all security architects and designers worldwide. All kinds of topics that help creating a security or privacy solution architecture are outlined, such as: security and privacy principles, common attack vectors, threat models while in-depth guidelines are also given to evaluate the use of Open Source security and privacy application in various use cases.

"Organizations worldwide rely on Java code to perform mission-critical tasks, and therefore that code must be reliable, robust, fast, maintainable, and secure. JavaTM Coding Guidelines brings together expert guidelines, recommendations, and code examples to help you meet these demands."--Publisher description.

The only comprehensive set of guidelines for secure Java programming - from the field's leading organizations, CERT and Oracle • •Authoritative, end-to-end code-level requirements for building secure systems with any recent version of Java, including the new Java 7 •Presents techniques that also improve safety, reliability, dependability, robustness, availability, maintainability, and other attributes of quality. •Includes extensive risk assessment guidance, plus references for further information. This is the first authoritative, comprehensive compilation of code-level requirements for building secure systems in Java. Organized by CERT's pioneering software security experts, with support from Oracle's own Java platform developers, it covers every facet of secure software coding with Java 7 SE and Java 6 SE, and offers value even to developers working with other Java versions. The authors itemize the most common coding errors leading to vulnerabilities in Java programs, and provide specific guidelines for avoiding each of them. They show how to produce programs that are not only secure, but also safer, more reliable, more robust, and easier to maintain. After a high-level introduction to Java application security, eighteen consistently-organized chapters detail specific guidelines for each facet of Java development. Each set of guidelines defines conformance, presents both noncompliant examples and corresponding compliant solutions, shows how to assess risk, and offers references for further information. To limit this book's size, the authors focus on 'normative requirements': strict rules for what programmers must do for their work to be secure, as defined by conformance to specific standards that can be tested through automated analysis software. (Note: A follow-up book will present 'non-normative requirements': recommendations for what Java developers typically 'should' do to further strengthen program security beyond testable 'requirements.')

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.

Consistent, high-quality coding standards improve software quality, reduce time-to-market, promote teamwork, eliminate time wasted on inconsequential matters, and simplify maintenance. Now, two of the world's most respected C++ experts distill the rich collective experience of the global C++ community into a set of coding standards that every developer and development team can understand and use as a basis for their own coding standards. The authors cover virtually every facet of C++ programming: design and coding style, functions, operators, class design, inheritance, construction/destruction, copying, assignment, namespaces, modules, templates, genericity, exceptions, STL containers and algorithms, and more. Each standard is described concisely, with practical examples. From type definition to error handling, this book presents C++ best practices, including some that have only

recently been identified and standardized-techniques you may not know even if you've used C++ for years. Along the way, you'll find answers to questions like What's worth standardizing--and what isn't? What are the best ways to code for scalability? What are the elements of a rational error handling policy? How (and why) do you avoid unnecessary initialization, cyclic, and definitional dependencies? When (and how) should you use static and dynamic polymorphism together? How do you practice "safe" overriding? When should you provide a no-fail swap? Why and how should you prevent exceptions from propagating across module boundaries? Why shouldn't you write namespace declarations or directives in a header file? Why should you use STL vector and string instead of arrays? How do you choose the right STL search or sort algorithm? What rules should you follow to ensure type-safe code? Whether you're working alone or with others, C++ Coding Standards will help you write cleaner code--and write it faster, with fewer hassles and less frustration.

The professional programmer’s Deitel® guide to procedural programming in C through 130 working code examples Written for programmers with a background in high-level language programming, this book applies the Deitel signature live-code approach to teaching the C language and the C Standard Library. The book presents the concepts in the context of fully tested programs, complete with syntax shading, code highlighting, code walkthroughs and program outputs. The book features approximately 5,000 lines of proven C code and hundreds of savvy tips that will help you build robust applications. Start with an introduction to C, then rapidly move on to more advanced topics, including building custom data structures, the Standard Library, select features of the new C11 standard such as multithreading to help you write high-performance applications for today’s multicore systems, and secure C programming sections that show you how to write software that is more robust and less vulnerable. You’ll enjoy the Deitels’ classic treatment of procedural programming. When you’re finished, you’ll have everything you need to start building industrial-strength C applications. Practical, example-rich coverage of: C programming fundamentals Compiling and debugging with GNU gcc and gdb, and Visual C++® Key new C11 standard features: Type generic expressions, anonymous structures and unions, memory alignment, enhanced Unicode® support, _Static_assert, quick_exit and at_quick_exit, _Noreturn function specifier, C11 headers C11 multithreading for enhanced performance on today’s multicore systems Secure C Programming sections Data structures, searching and sorting Order of evaluation issues, preprocessor Designated initializers, compound literals, bool type, complex numbers, variable-length arrays, restricted pointers, type generic math, inline functions, and more. Visit www.deitel.com For information on Deitel’s Dive Into® Series programming training courses delivered at organizations worldwide visit www.deitel.com/training or write to deitel@deitel.com Download code examples To receive updates for this book, subscribe to the free DEITEL® BUZZ ONLINE e-mail newsletter at www.deitel.com/newsletter/subscribe.html Join the Deitel social networking communities on Facebook® at facebook.com/DeitelFan , Twitter® @deitel, LinkedIn® at bit.ly/DeitelLinkedIn and Google+™ at gplus.to/Deitel