The CERT® C Coding Standard, Second Edition: 98 Rules For Developing Safe, Reliable, And Secure Systems (SEI Series In Software Engineering)

This Expert Guide gives you the techniques and technologies in software engineering to optimally design and implement your embedded system. Written by experts with a solutions focus, this encyclopedic reference gives you an indispensable aid to tackling the day-to-day problems when using software engineering methods to develop your embedded systems. With this book you will learn: The principles of good architecture for an embedded system Design practices to help make your embedded project successful Details on principles that are often a part of embedded systems, including digital signal processing, safety-critical principles, and development processes Techniques for setting up a performance engineering strategy for your embedded system software How to develop user interfaces for embedded systems Strategies for testing and deploying your embedded system, and ensuring quality development processes Practical techniques for optimizing embedded software for performance, memory, and power Advanced guidelines for developing multicore software for embedded systems How to develop embedded software for networking, storage, and automotive segments How to manage the embedded

development process Includes contributions from: Frank Schirrmeister, Shelly Gretlein, Bruce Douglass, Erich Styger, Gary Stringham, Jean Labrosse, Jim Trudeau, Mike Brogioli, Mark Pitchford, Catalin Dan Udma, Markus Levy, Pete Wilson, Whit Waldo, Inga Harris, Xinxin Yang, Srinivasa Addepalli, Andrew McKay, Mark Kraeling and Robert Oshana. Road map of key problems/issues and references to their solution in the text Review of core methods in the context of how to apply them Examples demonstrating timeless implementation details Short and to- the- point case studies show how key ideas can be implemented, the rationale for choices made, and design guidelines and trade-offs 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.

3+ Hours of Video Instruction Secure Coding Rules for Java: Serialization LiveLessons provides

developers with practical guidance for securely implementing Java Serialization. Overview Secure coding expert, Robert C. Seacord trains developers to understand Java serialization and the inherent security risks. Seacord also demonstrates how to securely implement serializable classes and evaluate mitigation strategies and alternative solutions. Java deserialization is an insecure language features that is widely used both directly by applications and indirectly by Java modules and libraries. Deserialization of untrusted streams can result in remote code execution (RCE), denial-of service (DoS), and a range of other exploits. Applications can be vulnerable to these attacks even when they are free from coding defects. Related Titles: Secure Coding Rules in Java: Part 1 LiveLessons (Video) The CERT Oracle Secure Coding Standard for Java (Book) Java Coding Guidelines: 75 Recommendations for Reliable and Secure Programs (Book) About the Instructor Robert C. Seacord is a Technical Director with NCC Group where he works with software developers and software development organizations to eliminate vulnerabilities resulting from coding errors before they are deployed. Previously, Robert led the secure coding initiative in the CERT Division of Carnegie Mellon University's Software Engineering Institute (SEI). Robert is also an adjunct professor in the School of Computer Science and the Information Networking Institute at Carnegie Mellon University. Robert is the author of six books, including The CERT C Coding Standard, Second Edition (Addison-Wesley, 2014), Secure Coding in C and C++, Second Edition (Addison-Wesley, 2013), The CERT Oracle Secure Coding Standard for Java

(Addison-Wesley, 2012), and Java Coding Guidelines: 75 Recommendations for Reliable and Secure Programs (Addison-Wesley, 2014). Robert is on the Advisory Board for the Linux Foundation and an expert on the ISO/IEC JTC1/SC22/WG14 international standardization working group for the C programming language. Skill Level Advanced Learning objectives: Understand Java object serialization Understand serialization security risks Understand deserialization vulnerabilities How to securely implement serializable classes Evaluate migration strategies Evaluate alternative solutions Who Should Take This Course Experienced Java developers Course Requirements Understanding of programming and development Expe...

"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.

A Confluence of Disciplines

Extreme C

Secure Programming with Python

National Cyber Summit (NCS) Research Track 2021

21st Century C

Intermediate 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.

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.

This text introduces the spirit and theory of hacking as well as the science behind it all; it also provides some core techniques and tricks of hacking so you can think like a hacker, write your own

hacks or thwart potential system attacks.

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.

Guidelines for the Use of the C Language in Critical Systems
Pro Cryptography and Cryptanalysis
Principles and Practices
Problem Solving And Program Design In C, 5/E
Evaluation of CERT Secure Coding Rules Through Integration with Source Code Analysis Tools
Hacking- The art Of Exploitation

This book comprises selected papers of the Third International Conference on Future Generation Information Technology, FGIT 2011, held in Jeju Island, Korea, in December 2011. The papers presented were carefully reviewed and selected from numerous submissions and focuse on the various aspects of advances in information technology. They were selected from the following 13 conferences: ASEA 2011, BSBT 2011, CA 2011, CES3 2011, DRBC 2011, DTA 2011, EL 2011, FGCN 2011, GDC 2011, MulGraB 2011, SecTech 2011, SIP 2011 and UNESST 2011.

"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. 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.

Push the limits of what C - and you - can do, with this high-intensity guide to the most advanced capabilities of C Key Features Make the most of C's low-level control,

flexibility, and high performance A comprehensive guide to C's most powerful and challenging features A thought-provoking guide packed with hands-on exercises and examples Book Description There's a lot more to C than knowing the language syntax. The industry looks for developers with a rigorous, scientific understanding of the principles and practices. Extreme C will teach you to use C's advanced low-level power to write effective, efficient systems. This intensive, practical guide will help you become an expert C programmer. Building on your existing C knowledge, you will master preprocessor directives, macros, conditional compilation, pointers, and much more. You will gain new insight into algorithm design, functions, and structures. You will discover how C helps you squeeze maximum performance out of critical, resource-constrained applications. C still plays a critical role in 21st-century programming, remaining the core language for precision engineering, aviations, space research, and more. This book shows how C works with Unix, how to implement OO principles in C, and fully covers multi-processing. In Extreme C, Amini encourages you to think, question, apply, and experiment for yourself. The book is essential for anybody who wants to take their C to the next level. What you will learn Build advanced C knowledge on strong foundations, rooted in first principles Understand memory structures and compilation

pipeline and how they work, and how to make most out of them Apply object-oriented design principles to your procedural C code Write low-level code that's close to the hardware and squeezes maximum performance out of a computer system Master concurrency, multithreading, multi-processing, and integration with other languages Unit Testing and debugging, build systems, and inter-process communication for C programming Who this book is for Extreme C is for C programmers who want to dig deep into the language and its capabilities. It will help you make the most of the low-level control C gives you.

Ten Strategies of a World-Class Cybersecurity Operations Center

C++ FAQs

The CERT Oracle Secure Coding Standard for Java

Best Practices for Development

101 Rules, Guidelines, and Best Practices

98 Rules for Developing Safe, Reliable, and Secure Systems

Utilize this comprehensive, yet practical, overview of modern cryptography and cryptanalysis to improve performance. Learn by example with source code in C# and .NET, and come away with an understanding of public key encryption systems and challenging cryptography mechanisms such as lattice-based cryptography. Modern cryptography is the lifeboat of a secure infrastructure. From global economies

and governments, to meeting everyday consumer needs, cryptography is ubiquitous, and used in search, design, data, artificial intelligence, and other fields of information technology and communications. Its complexity can lead to misconfiguration, misuse, and misconceptions. For developers who are involved in designing and implementing cryptographic operations in their applications, understanding the implications of the algorithms, modes, and other parameters is vital. Pro Cryptography and Cryptanalysis is for the reader who has a professional need or personal interest in developing cryptography algorithms and security schemes using C# and .NET. You will learn how to implement advanced cryptographic algorithms (such as Elliptic Curve Cryptography Algorithms, Lattice-based Cryptography, Searchable Encryption, Homomorphic Encryption), and come away with a solid understanding of the internal cryptographic mechanisms, and common ways in which the algorithms are correctly implemented in real practice. With the new era of quantum computing, this book serves as a stepping stone to quantum cryptography, finding useful connections between current cryptographic concepts and quantum related topics. What You Will Learn Know when to enlist cryptography, and how it is often misunderstood and misused Explore modern cryptography algorithms, practices, and properties

Design and implement usable, advanced cryptographic methods and mechanisms Understand how new features in C# and .NET impact the future of cryptographic algorithms Use the cryptographic model, services, and System.Security.Cryptography namespace in .NET Modernize your cryptanalyst mindset by exploiting the performance of C# and .NET with its weak cryptographic algorithms Practice the basics of public key cryptography, including ECDSA signatures Discover how most algorithms can be broken Who This Book Is For Information security experts, cryptologists, software engineers, developers, data scientists, and academia who have experience with C#, .NET, as well as IDEs such as Visual Studio, VS Code, or Mono. Because this book is for an intermediate to advanced audience, readers should also possess an understanding of cryptography (symmetric and asymmetric) concepts.

Ten Strategies of a World-Class Cyber Security Operations Center conveys MITRE's accumulated expertise on enterprise-grade computer network defense. It covers ten key qualities of leading Cyber Security Operations Centers (CSOCs), ranging from their structure and organization, to processes that best enable smooth operations, to approaches that extract maximum value from key CSOC technology investments. This book offers perspective and

context for key decision points in structuring a CSOC, such as what capabilities to offer, how to architect large-scale data collection and analysis, and how to prepare the CSOC team for agile, threat-based response. If you manage, work in, or are standing up a CSOC, this book is for you. It is also available on MITRE's website, www.mitre.org.

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.

Literate programming is a programming methodology that combines a programming language with a documentation language, making programs more easily maintained than programs written only in a high-level language. A literate

programmer is an essayist who writes programs for humans to understand. When programs are written in the recommended style they can be transformed into documents by a document compiler and into efficient code by an algebraic compiler. This anthology of essays includes Knuth's early papers on related topics such as structured programming as well as the Computer Journal article that launched literate programming. Many examples are given, including excerpts from the programs for TeX and METAFONT. The final essay is an example of CWEB, a system for literate programming in C and related languages. Index included.

The Hitchhiker's Guide to Python

Java Coding Guidelines

Secure Coding

The CERT® C Coding Standard, Second Edition

CERT C Secure Coding Standard

Fundamentals of Digital Signal Processing Using MATLAB

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.')

Throw out your old ideas of C, and relearn a programming language that's substantially outgrown its origins. With 21st Century C, you'll discover up-to-date techniques that are absent from every other C text available. C isn't just the foundation of modern programming languages, it is a modern language, ideal for writing efficient, state-of-the-art applications. Learn to dump old habits that made sense on

mainframes, and pick up the tools you need to use this evolved and aggressively simple language. No matter what programming language you currently champion, you'll agree that C rocks. Set up a C programming environment with shell facilities, makefiles, text editors, debuggers, and memory checkers Use Autotools, C's de facto cross-platform package manager Learn which older C concepts should be downplayed or deprecated Explore problematic C concepts that are too useful to throw out Solve C's string-building problems with C-standard and POSIX-standard functions Use modern syntactic features for functions that take structured inputs Build high-level object-based libraries and programs Apply existing C libraries for doing advanced math, talking to Internet servers, and running databases

Discover How Electronic Health Records Are Built to Drive the Next Generation of Healthcare Delivery The increased role of IT in the healthcare sector has led to the coining of a new phrase "health informatics," which deals with the use of IT for better healthcare

services. Health informatics applications often involve maintaining the health records of individuals, in digital form, which is referred to as an Electronic Health Record (EHR). Building and implementing an EHR infrastructure requires an understanding of healthcare standards, coding systems, and frameworks. This book provides an overview of different health informatics resources and artifacts that underlie the design and development of interoperable healthcare systems and applications. Electronic Health Record: Standards, Coding Systems, Frameworks, and Infrastructures compiles, for the first time, study and analysis results that EHR professionals previously had to gather from multiple sources. It benefits readers by giving them an understanding of what roles a particular healthcare standard, code, or framework plays in EHR design and overall IT-enabled healthcare services along with the issues involved. This book on Electronic Health Record: Offers the most comprehensive coverage of available EHR Standards including

ISO, European Union Standards, and national initiatives by Sweden, the Netherlands, Canada, Australia, and many others Provides assessment of existing standards Includes a glossary of frequently used terms in the area of EHR Contains numerous diagrams and illustrations to facilitate comprehension Discusses security and reliability of data

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.

Third International Conference, FGIT 2011, Jeju Island, December 8-10, 2011. Proceedings

Electronic Health Record

MISRA-C:2004

Embedded C Coding Standard

Modern C

Software Engineering for Embedded Systems

The Hitchhiker's Guide to Python takes the journeyman Pythonista to true expertise. More than any other language, Python was created with the philosophy of simplicity and parsimony. Now 25 years old, Python has become the primary or secondary language (after SQL) for many business users. With popularity comes diversity—and possibly dilution. This guide, collaboratively written by over a hundred members of the Python community, describes best practices currently used by package and application developers. Unlike other books for this audience, The Hitchhiker's Guide is light on reusable code and heavier on design philosophy, directing the reader to excellent sources that already exist.

"A must-read for all Java developers. . . . Every developer has a responsibility to author code that is free of significant security vulnerabilities. This book provides realistic guidance to help Java developers implement desired functionality with security, reliability,

and maintainability goals in mind." –Mary Ann Davidson, Chief Security Officer, Oracle Corporation Organizations worldwide rely on Java code to perform mission-critical tasks, and therefore that code must be reliable, robust, fast, maintainable, and secure. Java™ Coding Guidelines brings together expert guidelines, recommendations, and code examples to help you meet these demands. Written by the same team that brought you The CERT® Oracle ® Secure Coding Standard for Java™, this guide extends that previous work's expert security advice to address many additional quality attributes. You'll find 75 guidelines, each presented consistently and intuitively. For each guideline, conformance requirements are specified; for most, noncompliant code examples and compliant solutions are also offered. The authors explain when to apply each guideline and provide references to even more detailed information. Reflecting pioneering research on Java security, Java™ Coding Guidelines offers updated techniques for protecting against both deliberate attacks and other unexpected events. You'll find best practices for improving code reliability and clarity, and a full chapter exposing common misunderstandings that lead to suboptimal code. With a Foreword by James A. Gosling, Father of the Java Programming Language

This second edition text focuses on the fundamentals of digital signal processing with an emphasis on practical applications. In order to motivate students, many of the examples illustrate the processing of speech and music. This theme is also a focus of the course software that features facilities for recording and playing sound on a standard PC. The accompanying website contains a comprehensive MATLAB software package called the Fundamentals of Digital Signal Processing (FDSP) toolbox version 2.0. The FDSP toolbox includes chapter GUI modules, an extensive library of DSP functions, direct access to all of the computational examples, figures, and tables, solutions to selected problems, and onliine help documentation. Using the interactive GUI modules, students can explore, compare, and directly experience the effects of signal processing techniques without any need for programming. Important Notice: Media content

referenced within the product description or the product text may not be available in the ebook version.

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.

Eh

Creating Advanced Algorithms with C# and .NET

With C and GNU Development Tools

The CERT C Coding Standard

MISRA-C: 2012

Enterprise Software Security

STRENGTHEN SOFTWARE SECURITY BY HELPING DEVELOPERS AND SECURITY EXPERTS WORK TOGETHER Traditional approaches to securing software are inadequate. The solution: Bring software engineering and network security teams together in a new, holistic approach to protecting the entire enterprise. Now, four highly respected security experts explain why this “confluence” is so crucial, and show how to implement it in your organization. Writing for all software and security practitioners and leaders, they show how software can play a vital,

active role in protecting your organization. You'll learn how to construct software that actively safeguards sensitive data and business processes and contributes to intrusion detection/response in sophisticated new ways. The authors cover the entire development lifecycle, including project inception, design, implementation, testing, deployment, operation, and maintenance. They also provide a full chapter of advice specifically for Chief Information Security Officers and other enterprise security executives. Whatever your software security responsibilities, Enterprise Software Security delivers indispensable big-picture guidance–and specific, high-value recommendations you can apply right now. COVERAGE INCLUDES: • Overcoming common obstacles to collaboration between developers and IT security professionals • Helping programmers design, write, deploy, and operate more secure software • Helping network security engineers use application output more effectively • Organizing a software security team before you've even created requirements • Avoiding the unmanageable complexity and inherent flaws of layered security • Implementing positive software design practices and identifying security defects in existing designs • Teaming to improve code reviews, clarify attack scenarios associated with vulnerable code, and validate positive compliance • Moving beyond pentesting toward more comprehensive security testing • Integrating your new application with your existing security infrastructure • "Ruggedizing" DevOps by adding infosec to the relationship between development and operations • Protecting application security during maintenance

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.

"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

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.

Effective C

C++ Primer Plus

Methods, Practical Techniques, and Applications

C++ Coding Standards

Serialization

Secure Coding Rules for Java

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.
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.

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.

The CERT C Secure Coding StandardAddison-Wesley Professional

Programming Embedded Systems

Future Generation Information Technology

75 Recommendations for Reliable and Secure Programs

Secure Coding in C and C++

Standards, Coding Systems, Frameworks, and Infrastructures

Secure Programming Cookbook for C and C++

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."

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.

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

This book presents findings from the papers accepted at the Cyber Security Education Stream and Cyber Security Technology Stream of The National Cyber Summit's Research Track, reporting on latest advances on topics ranging from software security to cyber-attack detection and modelling to the use

of machine learning in cyber security to legislation and policy to surveying of small businesses to cyber competition, and so on. Understanding the latest capabilities in cyber security ensures users and organizations are best prepared for potential negative events. This book is of interest to cyber security researchers, educators and practitioners, as well as students seeking to learn about cyber security.

Literate Programming

The CERT C Secure Coding Standard

C Tips from the New School

Taking you to the limit in Concurrency, OOP, and the most advanced capabilities of C

An Introduction to Professional C Programming

Recipes for Cryptography, Authentication, Input Validation & More