The Art Of Unit Testing: With Examples In C

The Art of Unit Testing

Summary The Art of Unit Testing, Second Edition guides you step by step from writing your first simple tests to developing robust test sets that are maintainable, readable, and trustworthy. You'll master the foundational ideas and quickly move to high-value subjects like mocks, stubs, and isolation, including frameworks such as Moq, FakeItEasy, and Typemock Isolator. You'll explore test patterns and organization, working with legacy code, and even \"untestable\" code. Along the way, you'll learn about integration testing and techniques and tools for testing databases and other technologies. About this Book You know you should be unit testing, so why aren't you doing it? If you're new to unit testing, if you find unit testing tedious, or if you're just not getting enough payoff for the effort you put into it, keep reading. The Art of Unit Testing, Second Edition guides you step by step from writing your first simple unit tests to building complete test sets that are maintainable, readable, and trustworthy. You'll move quickly to more complicated subjects like mocks and stubs, while learning to use isolation (mocking) frameworks like Moq, FakeItEasy, and Typemock Isolator. You'll explore test patterns and organization, refactor code applications, and learn how to test \"untestable\" code. Along the way, you'll learn about integration testing and techniques for testing with databases. The examples in the book use C#, but will benefit anyone using a statically typed language such as Java or C++. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. What's Inside Create readable, maintainable, trustworthy tests Fakes, stubs, mock objects, and isolation (mocking) frameworks Simple dependency injection techniques Refactoring legacy code About the Author Roy Osherove has been coding for over 15 years, and he consults and trains teams worldwide on the gentle art of unit testing and test-driven development. His blog is at ArtOfUnitTesting.com. Table of Contents PART 1 GETTING STARTED The basics of unit testing A first unit test PART 2 CORE TECHNIQUES Using stubs to break dependencies Interaction testing using mock objects Isolation (mocking) frameworks Digging deeper into isolation frameworks PART 3 THE TEST CODE Test hierarchies and organization The pillars of good unit tests PART 4 DESIGN AND PROCESS Integrating unit testing into the organization Working with legacy code Design and testability

The Art of Unit Testing, Third Edition

The art of unit testing is more than just learning the right collection of tools and practices. It's about understanding what makes great tests tick, finding the right strategy for each unique situation, and knowing what to do when the testing process gets messy. This book delivers insights and advice that will transform the way you test your software. The art of unit testing, third edition shows you how to create readable and maintainable tests. It goes well beyond basic test creation into organization-wide test strategies, troubleshooting, working with legacy code, and \"merciless\" refactoring. You'll love the practical examples and familiar scenarios that make testing come alive as you read. This third edition has been updated with techniques specific to object-oriented, functional, and modular coding styles. The examples use JavaScript.

The Art of Unit Testing, Third Edition

Unit testing is more than just a collection of tools and practices—it's a state of mind! This bestseller reveals the master's secrets for delivering robust, maintainable, and trustworthy code. Thousands of developers have learned to hone their code quality under the tutelage of The Art of Unit Testing. This revised third edition updates an international bestseller to reflect modern development tools and practices, as well as to cover JavaScript. Inside The Art of Unit Testing, Third Edition you will learn how to: Create readable, maintainable, and trustworthy tests Work with fakes, stubs, mock objects, and isolation frameworks Apply simple dependency injection techniques Refactor legacy code with confidence Test both frontend and backend code Effective unit tests streamline your software development process and ensure you deliver consistent high-quality code every time. With practical examples in JavaScript and Node, this hands-on guide takes you from your very first unit tests all the way to comprehensive test suites, naming standards, and refactoring techniques. You'll explore test patterns and organization, working with legacy code and even "untestable" code. The many tool-agnostic examples are presented in JavaScript and carefully designed so that they apply to code written in any language. About the technology The art of unit testing is more than just learning the right collection of tools and practices. It's about understanding what makes great tests tick, finding the right strategy for each unique situation, and knowing what to do when the testing process gets messy. This book delivers insights and advice that will transform the way you test your software. About the book The Art of Unit Testing, Third Edition shows you how to create readable and maintainable tests. It goes well beyond basic test creation into organization-wide test strategies, troubleshooting, working with legacy code, and "merciless" refactoring. You'll love the practical examples and familiar scenarios that make testing come alive as you read. This third edition has been updated with techniques specific to object-oriented, functional, and modular coding styles. The examples use JavaScript. What's inside Deciding on test types and strategies Test Entry & Exit Points Refactoring legacy code Fakes, stubs, mock objects, and isolation frameworks Object-Oriented, Functional, and Modular testing styles About the reader Examples use JavaScript, TypeScript, and Node.js. About the author Roy Osherove is an internationally-recognized expert in unit testing and agile software methodology. Vladimir Khorikov is the author of Manning's Unit Testing Principles, Practices, and Patterns, a Pluralsight author, and a Microsoft MVP. Table of Contents PART 1 1 The basics of unit testing 2 A first unit test PART 2 3 Breaking dependencies with stubs 4 Interaction testing using mock objects 5 Isolation frameworks 6 Unit testing asynchronous code PART 3 7 Trustworthy tests 8 Maintainability PART 4 9 Readability 10 Developing a testing strategy 11 Integrating unit testing into the organization 12 Working with legacy code Appendix Monkey-patching functions and modules

Mastering the Art of Unit Testing: Unraveling the Secrets of Expert-Level Programming

Dive deep into the precision-driven world of software quality with \"Mastering the Art of Unit Testing: Unraveling the Secrets of Expert-Level Programming.\" This essential guide equips seasoned developers with the advanced strategies and insights necessary to refine their unit testing practices. Carefully curated to explore the intricate facets of unit testing, this book provides a detailed exploration of foundational principles, sophisticated design patterns, and the integration of test automation within Agile and DevOps environments. Each chapter meticulously covers crucial aspects, ranging from effective usage of mocks and stubs to mastering Test-Driven Development and its seamless integration with refactoring. Readers will gain expertise in handling legacy code challenges, ensuring comprehensive test coverage, and utilizing cutting-edge automated testing frameworks. By bringing clarity to complex topics like asynchronous code testing and future trends influenced by AI and machine learning, this book becomes an indispensable resource for maintaining software quality and adaptability. \"Mastering the Art of Unit Testing\" isn't just a guide—it's a transformative toolkit designed to elevate your testing capabilities and deepen your understanding of contemporary testing methodologies. Whether you're aiming to bolster your project's test reliability or embrace future testing innovations, this book offers the practical wisdom and expert knowledge needed to achieve excellence in software development. Join the ranks of expert programmers committed to advancing their craft and ensuring superior software quality.

Applied Computer Science for GGOS Observatories

This book combines elementary theory from computer science with real-world challenges in global geodetic observation, based on examples from the Geodetic Observatory Wettzell, Germany. It starts with a step-by-step introduction to developing stable and safe scientific software to run successful software projects. The use of software toolboxes is another essential aspect that leads to the application of generative programming. An example is a generative network middleware that simplifies communication. One of the book's main focuses is on explaining a potential strategy involving autonomous production cells for space geodetic techniques. The complete software design of a satellite laser ranging system is taken as an example. Such automated systems are then combined for global interaction using secure communication tunnels for remote access. The network of radio telescopes is used as a reference. Combined observatories form coordinated multi-agent systems and offer solutions for operational aspects of the Global Geodetic Observing System (GGOS) with regard to “Industry 4.0”.

iOS Unit Testing by Example

Fearlessly change the design of your iOS code with solid unit tests. Use Xcode's built-in test framework XCTest and Swift to get rapid feedback on all your code - including legacy code. Learn the tricks and techniques of testing all iOS code, especially view controllers (UIViewControllers), which are critical to iOS apps. Learn to isolate and replace dependencies in legacy code written without tests. Practice safe refactoring that makes these tests possible, and watch all your changes get verified quickly and automatically. Make even the boldest code changes with complete confidence. Manual code and UI testing get slower the deeper your navigation hierarchy goes. It can take several taps just to reach a particular screen, never mind the actual workflow tests. Automatic unit testing offers such rapid feedback that it can change the rules of development. Bring testing to iOS development, even for legacy code. Use XCTest to write unit tests in Swift for all your code. iOS developers typically reserve unit tests for their model classes alone. But that approach skips most of the code common to iOS apps, especially with UIViewControllers. Learn how to unit test these view controllers to expand your unit testing possibilities. Since good unit tests form the bedrock for safe refactoring, you're empowered to make bold changes. Learn how to avoid the most common mistakes Swift programmers make with the XCTest framework. Use code coverage to find holes in your test suites. Learn how to identify hard dependencies. Reshape the design of your code quickly, with less risk and less fear.

Unity Cookbook

Are you ready to take your Unity game development skills to the next level? Look no further! The \"Unity Cookbook 2023, 5th Edition\" is your essential guide to mastering the latest features of Unity 2023, packed with over 140 recipes to empower your game development journey. Purchase of the print or Kindle book includes a free eBook in the PDF format. Key Features Explore VR and AR development to create immersive experiences that redefine gaming Craft captivating mobile games with optimized performance and user-friendly controls Elevate gameplay with expertly composed music, dynamic sound effects, and seamless audio integration Book DescriptionUnleash your game development potential with Unity Cookbook, 5th Edition, designed to equip you with the skills and knowledge needed to excel in Unity game development. With over 160 expertly crafted recipes empowering you to pioneer VR and AR experiences, excel in mobile game development, and become a master of audio techniques. In this latest edition, we've meticulously curated a collection of recipes that reflect the latest advancements in Unity 2023, ensuring you stay at the forefront of game development. You'll discover dedicated recipes for First/Third Person (Core) templates, create engaging mobile games, delve into Virtual and Augmented Reality, and go further with audio by exploring advanced techniques. Additionally, the book has been fully updated to incorporate the new input system and TextMeshPro, essential elements for modern game development. From exploring C# scripting to crafting stylish UIs, creating stunning visual effects, and understanding shader development through Shader Graph, every chapter is designed to take you closer to your goal of becoming a proficient Unity developer. So, whether you're aiming to develop the next hit game, enhance your portfolio, or simply have fun building games, this book will be your trusted companion on your journey to Unity proficiency.What you will learn Craft stylish user interfaces, from power bars to radars, and implement button-driven scene changes effortlessly Enhance your games with AI controlled characters, harnessing Unity's navigation meshes, surfaces, and agents Discover the power of Cinemachine in Unity for intelligent camera movements Elevate games with immersive audio, including background music and dynamic sound effects Bring your games to life with captivating visual effects, from smoke and explosions to customizable particle systems Build your own shaders using Unity's Shader Graph tool Who this book is for If you’re a Unity developer looking for better ways to resolve common recurring problems, then this book is for you. Programmers dipping their toes into multimedia features for the first time will also find this book useful. Before you get started with this book, you’ll need a solid understanding of Unity’s functionality and experience with programming in C#.

The Coding Dojo Handbook

This handbook is a collection of concrete ideas for how you can get started with a Coding Dojo, where a group of programmers can focus on improving their practical coding skills.

Data Clustering in C++

Data clustering is a highly interdisciplinary field, the goal of which is to divide a set of objects into homogeneous groups such that objects in the same group are similar and objects in different groups are quite distinct. Thousands of theoretical papers and a number of books on data clustering have been published over the past 50 years. However, few books exist to teach people how to implement data clustering algorithms. This book was written for anyone who wants to implement or improve their data clustering algorithms. Using object-oriented design and programming techniques, Data Clustering in C++ exploits the commonalities of all data clustering algorithms to create a flexible set of reusable classes that simplifies the implementation of any data clustering algorithm. Readers can follow the development of the base data clustering classes and several popular data clustering algorithms. Additional topics such as data pre-processing, data visualization, cluster visualization, and cluster interpretation are briefly covered. This book is divided into three parts-- Data Clustering and C++ Preliminaries: A review of basic concepts of data clustering, the unified modeling language, object-oriented programming in C++, and design patterns A C++ Data Clustering Framework: The development of data clustering base classes Data Clustering Algorithms: The implementation of several popular data clustering algorithms A key to learning a clustering algorithm is to implement and experiment the clustering algorithm. Complete listings of classes, examples, unit test cases, and GNU configuration files are included in the appendices of this book as well as in the downloadable resources. The only requirements to compile the code are a modern C++ compiler and the Boost C++ libraries.

The Art of C Programming

Unlock the power of C programming with \"The Art of C Programming,\" your essential guide to mastering dynamic memory management. Delve into the depths of this critical aspect of C programming and elevate your coding skills to new heights. Begin your journey with dynamic memory allocation, where you'll explore memory management strategies that breathe life into your applications. Discover the intricacies of pointers and their dynamic capabilities, learning to manipulate memory efficiently using functions like `malloc()` and `calloc()`. Master the art of releasing memory with `free()` and adhere to best practices that ensure your programs run smoothly and without leaks. Advance into sophisticated pointer techniques, where you'll harness the power of pointers to pointers, arrays, and function pointers. Engage in generic programming through void pointers, pushing the boundaries of what your programs can achieve. Transform your coding arsenal with data structures powered by dynamic memory. Implement linked lists, dynamic arrays, stacks, queues, and dynamic hash tables. These structures will offer your applications unparalleled flexibility and responsiveness. Explore string manipulation within dynamic memory, ensuring your programs handle character data safely and efficiently. Learn to identify memory leaks and utilize powerful tools like Valgrind for memory analysis, avoiding common pitfalls and optimizing every byte of your program. Venture into the real-world applications of dynamic memory, from building memory-efficient applications to implementing microservices and excelling in embedded systems. Discover the synergy between dynamic memory and multithreading, threading the needle between memory management and concurrent programming. Deepen your understanding with chapters on optimizing code, integrating with external libraries, writing adaptable C code, security implications, and rigorous testing methodologies. Expand your knowledge further by linking dynamic memory concepts with other languages and developing dynamic APIs in C. \"The Art of C Programming\" is your comprehensive companion in navigating the challenges and opportunities of dynamic memory, empowering you to craft robust, scalable, and secure C applications. Embrace the art today!

Theory and Engineering of Dependable Computer Systems and Networks

This book contains papers on selected aspects of dependability analysis in computer systems and networks, which were chosen for discussion during the 16th DepCoS-RELCOMEX conference held in Wroc?aw, Poland, from June 28 to July 2, 2021. Their collection will be a valuable source material for scientists, researchers, practitioners and students who are dealing with design, analysis and engineering of computer systems and networks and must ensure their dependable operation. Being probably the most complex technical systems ever engineered by man (and also—the most dynamically evolving ones), organization of contemporary computer systems cannot be interpreted only as structures built on the basis of (unreliable) technical resources. Their evaluation must take into account a specific blend of interacting people (their needs and behaviours), networks (together with mobile properties, cloud organization, Internet of Everything, etc.) and a large number of users dispersed geographically and constantly producing an unconceivable number of applications. Ever-growing number of research methods being continuously developed for dependability analyses apply the newest techniques of artificial and computational intelligence. Selection of papers in these proceedings illustrates diversity of multi-disciplinary topics which are considered in present-day dependability explorations.

Effective Software Testing

Go beyond basic testing! Great software testing makes the entire development process more efficient. This book reveals a systemic and effective approach that will help you customize your testing coverage and catch bugs in tricky corner cases. In Effective Software Testing you will learn how to: Engineer tests with a much higher chance of finding bugs Read code coverage metrics and use them to improve your test suite Understand when to use unit tests, integration tests, and system tests Use mocks and stubs to simplify your unit testing Think of pre-conditions, post-conditions, invariants, and contracts Implement property-based tests Utilize coding practices like dependency injection and hexagonal architecture that make your software easier to test Write good and maintainable test code Effective Software Testing teaches you a systematic approach to software testing that will ensure the quality of your code. It’s full of techniques drawn from proven research in software engineering, and each chapter puts a new technique into practice. Follow the real-world use cases and detailed code samples, and you’ll soon be engineering tests that find bugs in edge cases and parts of code you’d never think of testing! Along the way, you’ll develop an intuition for testing that can save years of learning by trial and error. About the technology Effective testing ensures that you’ll deliver quality software. For software engineers, testing is a key part of the development process. Mastering specification-based testing, boundary testing, structural testing, and other core strategies is essential to writing good tests and catching bugs before they hit production. About the book Effective Software Testing is a hands-on guide to creating bug-free software. Written for developers, it guides you through all the different types of testing, from single units up to entire components. You’ll also learn how to engineer code that facilitates testing and how to write easy-to-maintain test code. Offering a thorough, systematic approach, this book includes annotated source code samples, realistic scenarios, and reasoned explanations. What's inside Design rigorous test suites that actually find bugs When to use unit tests, integration tests, and system tests Pre-and post-conditions, invariants, contracts, and property-based tests Design systems that are test-friendly Test code best practices and test smells About the reader The Java-based examples illustrate concepts you can use for any object-oriented language. About the author Dr. Maurício Aniche is the Tech Academy Lead at Adyen and an Assistant Professor in Software Engineering at the Delft University of Technology. Table of Contents 1 Effective and systematic software testing 2 Specification-based testing 3 Structural testing and code coverage 4 Designing contracts 5 Property-based testing 6 Test doubles and mocks 7 Designing for testability 8 Test-driven development 9 Writing larger tests 10 Test code quality 11 Wrapping up the book

ASP.NET Core in Action, Third Edition

ASP.NET Core in Action, 3rd edition by Microsoft MVP Andrew Lock, is a fully updated edition, ideal for intermediate C# developers, and teaches you how to use your C# and .NET skills to build amazing cross-platform web applications. Even if you have never worked with ASP.NET, you will soon start creating productive cross-platform web apps.

The Art of Lean Software Development

This succinct book explains how you can apply the practices of Lean software development to dramatically increase productivity and quality. Based on techniques that revolutionized Japanese manufacturing, Lean principles are being applied successfully to product design, engineering, the supply chain, and now software development. With The Art of Lean Software Development, you'll learn how to adopt Lean practices one at a time rather than taking on the entire methodology at once. As you master each practice, you'll see significant, measurable results. With this book, you will: Understand Lean's origins from Japanese industries and how it applies to software development Learn the Lean software development principles and the five most important practices in detail Distinguish between the Lean and Agile methodologies and understand their similarities and differences Determine which Lean principles you should adopt first, and how you can gradually incorporate more of the methodology into your process Review hands-on practices, including descriptions, benefits, trade-offs, and roadblocks Learn how to sell these principles to management The Art of Lean Software Development is ideal for busy people who want to improve the development process but can't afford the disruption of a sudden and complete transformation. The Lean approach has been yielding dramatic results for decades, and with this book, you can make incremental changes that will produce immediate benefits. \"This book presents Lean practices in a clear and concise manner so readers are motivated to make their software more reliable and less costly to maintain. I recommend it to anyone looking for an easy-to-follow guide to transform how the developer views the process of writing good software.\"-- Bryan Wells, Boeing Intelligence & Security Sytems Mission System \"If you're new to Lean software development and you're not quite sure where to start, this book will help get your development process going in the right direction, one step at a time.\"-- John McClenning, software development lead, Aclara

ASP.NET Core in Action, Second Edition

ASP.NET Core in Action, Second Edition is a comprehensive guide to creating web applications with ASP.NET Core 5.0. Go from basic HTTP concepts to advanced framework customization. Summary Fully updated to ASP.NET 5.0, ASP.NET Core in Action, Second Edition is a hands-on primer to building cross-platform web applications with your C# and .NET skills. Even if you've never worked with ASP.NET you'll start creating productive cross-platform web apps fast. And don’t worry about late-breaking changes to ASP.NET Core. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the technology Build full-stack web applications that run anywhere. Developers love ASP.NET Core for its libraries and pre-built components that maximize productivity. Version 5.0 offers new features for server-side apps, as well as background services for cross-platform development. About the book ASP.NET Core in Action, Second Edition is a comprehensive guide to creating web applications with ASP.NET Core 5.0. Go from basic HTTP concepts to advanced framework customization. Illustrations and annotated code make learning visual and easy. Master logins, dependency injection, security, and more. This updated edition covers the latest features, including Razor Pages and the new hosting paradigm. What's inside Developing apps for Windows and non-Windows servers Configuring applications Building custom components Logging, testing, and security About the reader For intermediate C# developers. About the author Andrew Lock is a Microsoft MVP who has worked with ASP.NET Core since before its first release. Table of Contents PART 1 - GETTING STARTED WITH ASP.NET CORE 1 Getting started with ASP.NET Core 2 Your first application 3 Handling requests with the middleware pipeline 4 Creating a website with Razor Pages 5 Mapping URLs to Razor Pages using routing 6 The binding model: Retrieving and validating user input 7 Rendering HTML using Razor views 8 Building forms with Tag Helpers 9 Creating a Web API for mobile and client applications using MVC PART 2 - BUILDING

COMPLETE APPLICATIONS 10 Service configuration with dependency injection 11 Configuring an ASP.NET Core application 12 Saving data with Entity Framework Core 13 The MVC and Razor Pages filter pipeline 14 Authentication: Adding users to your application with Identity 15 Authorization: Securing your application 16 Publishing and deploying your application PART 3 - EXTENDING YOUR APPLICATIONS 17 Monitoring and troubleshooting errors with logging 18 Improving your application's security 19 Building custom components 20 Building custom MVC and Razor Pages components 21 Calling remote APIs with IHttpClientFactory 22 Building background tasks and services 23 Testing your application

Artificial Intelligence Methods for Optimization of the Software Testing Process

Artificial Intelligence Methods for Optimization of the Software Testing Process: With Practical Examples and Exercises presents different AI-based solutions for overcoming the uncertainty found in many initial testing problems. The concept of intelligent decision making is presented as a multi-criteria, multi-objective undertaking. The book provides guidelines on how to manage diverse types of uncertainty with intelligent decision-making that can help subject matter experts in many industries improve various processes in a more efficient way. As the number of required test cases for testing a product can be large (in industry more than 10,000 test cases are usually created). Executing all these test cases without any particular order can impact the results of the test execution, hence this book fills the need for a comprehensive resource on the topics on the how's, what's and whys. To learn more about Elsevier's Series, Uncertainty, Computational Techniques and Decision Intelligence, please visit this link: https://www.elsevier.com/books-and-journals/book-series/uncertainty-computational-techniques-and-decision-intelligence - Presents one of the first empirical studies in the field, contrasting theoretical assumptions on innovations in a real industrial environment with a large set of use cases from developed and developing testing processes at various large industries - Explores specific comparative methodologies, focusing on developed and developing AI-based solutions - Serves as a guideline for conducting industrial research in the artificial intelligence and software testing domain - Explains all proposed solutions through real industrial case studies

Clean Code with C#

Enhance your programming skills through code reviews, TDD and BDD implementation, and API design to overcome code inefficiency, redundancy, and other issues arising from bad code Key Features Write code that seamlessly integrates with other systems while maintaining well-defined software boundaries Understand how coding principles and standards elevate software quality Learn how to avoid common errors while implementing concurrency or threading Purchase of the print or Kindle book includes a free PDF eBook Book DescriptionTraditionally associated with Windows desktop applications and game development, C# has expanded into web, cloud, and mobile development. However, despite its extensive coding features, professionals often encounter issues with efficiency, scalability, and maintainability due to poor code. Clean Code in C# guides you in identifying and resolving these problems using coding best practices. This book starts by comparing good and bad code to emphasize the importance of coding standards, principles, and methodologies. It then covers code reviews, unit testing, and test-driven development, and addresses cross-cutting concerns. As you advance through the chapters, you'll discover programming best practices for objects, data structures, exception handling, and other aspects of writing C# computer programs. You'll also explore API design and code quality enhancement tools, while studying examples of poor coding practices to understand what to avoid. By the end of this clean code book, you'll have the developed the skills needed to apply industry-approved coding practices to write clean, readable, extendable, and maintainable C# code.What you will learn Master the art of writing evolvable and adaptable code Implement the fail-pass-refactor methodology using a sample C# console application Develop custom C# exceptions that provide meaningful information Identify low-quality C# code in need of refactoring Improve code performance using profiling and refactoring tools Create efficient and bug-free code using functional programming techniques Write cross-platform code using MAUI Develop cloud-deployable microservices for versatile applications Who this book is for This coding book is for proficient C# developers, team leads, senior software engineers, and software architects who want to improve the efficiency of their legacy systems. A strong understanding of C# programming is assumed.

Developer Testing

How do successful agile teams deliver bug-free, maintainable software—iteration after iteration? The answer is: By seamlessly combining development and testing. On such teams, the developers write testable code that enables them to verify it using various types of automated tests. This approach keeps regressions at bay and prevents “testing crunches”—which otherwise may occur near the end of an iteration—from ever happening. Writing testable code, however, is often difficult, because it requires knowledge and skills that cut across multiple disciplines. In Developer Testing, leading test expert and mentor Alexander Tarlinder presents concise, focused guidance for making new and legacy code far more testable. Tarlinder helps you answer questions like: When have I tested this enough? How many tests do I need to write? What should my tests verify? You’ll learn how to design for testability and utilize techniques like refactoring, dependency breaking, unit testing, data-driven testing, and test-driven development to achieve the highest possible confidence in your software. Through practical examples in Java, C#, Groovy, and Ruby, you’ll discover what works—and what doesn’t. You can quickly begin using Tarlinder’s technology-agnostic insights with most languages and toolsets while not getting buried in specialist details. The author helps you adapt your current programming style for testability, make a testing mindset “second nature,” improve your code, and enrich your day-to-day experience as a software professional. With this guide, you will Understand the discipline and vocabulary of testing from the developer’s standpoint Base developer tests on well-established testing techniques and best practices Recognize code constructs that impact testability Effectively name, organize, and execute unit tests Master the essentials of classic and “mockist-style” TDD Leverage test doubles with or without mocking frameworks Capture the benefits of programming by contract, even without runtime support for contracts Take control of dependencies between classes, components, layers, and tiers Handle combinatorial explosions of test cases, or scenarios requiring many similar tests Manage code duplication when it can’t be eliminated Actively maintain and improve your test suites Perform more advanced tests at the integration, system, and end-to-end levels Develop an understanding for how the organizational context influences quality assurance Establish well-balanced and effective testing strategies suitable for agile teams

Springer Handbook of Engineering Statistics

In today’s global and highly competitive environment, continuous improvement in the processes and products of any field of engineering is essential for survival. This book gathers together the full range of statistical techniques required by engineers from all fields. It will assist them to gain sensible statistical feedback on how their processes or products are functioning and to give them realistic predictions of how these could be improved. The handbook will be essential reading for all engineers and engineering-connected managers who are serious about keeping their methods and products at the cutting edge of quality and competitiveness.

Practical Enterprise Software Development Techniques

This expanded and updated edition of \"Practical Enterprise Software Development Techniques\" includes a new chapter which explains what makes enterprise scale software development different from other development endeavors. Chapter 4 has been expanded with additional coverage of code review, bug tracker systems and agile software applications. The chapter order has been changed in response to feedback from readers and instructors who have taught classes using the previous version (which was also published by Apress). This book provides an overview of tools and techniques used in enterprise software development, many of which are not taught in academic programs or learned on the job. This is an ideal resource containing lots of practical information and code examples that you need to master as a member of an enterprise development team. This book aggregates many of these \"on the job\" tools and techniques into a concise format and presents them as both discussion topics andwith code examples. The reader will not only get an overview of these tools and techniques, but also several discussions concerning operational aspects of enterprise software development and how it differs from smaller development efforts. For example, in the chapter on Design Patterns and Architecture, the author describes the basics of design patterns but only highlights those that are more important in enterprise applications due to separation of duties, enterprise security, etc. The architecture discussion revolves has a similar emphasis – different teams may manage different aspects of the application's components with little or no access to the developer. This aspect of restricted access is also mentioned in the section on logging. Theory of logging and discussions of what to log are briefly mentioned, the configuration of the logging tools is demonstrated along with a discussion of why it's very important in an enterprise environment.

Careers in Focus: Computer and Video Game Design, Third Edition

Ferguson's Careers in Focus books are a valuable career exploration tool for libraries and career centers. Written in an easy-to-understand yet informative style, this series surveys a wide array of commonly held jobs and is arranged into volumes organized by specific industries and interests. Each of these informative books is loaded with up-to-date career information presented in a featured industry article and a selection of detailed professions articles. The information here has been researched, vetted, and analyzed by Ferguson's editors, drawing from government and industry sources, professional groups, news reports, career and job-search resources, and a variety of other sources. For readers making career choices, these books offer a wealth of helpful information and resources.

The Art of Effective C++: Building Robust Software with Precision

This comprehensive guide to C++ programming will equip you with the knowledge and skills you need to create robust, maintainable software applications. Whether you are a beginner or an experienced programmer, this book will take you from the basics of C++ to advanced concepts and techniques. With clear explanations, hands-on examples, and in-depth coverage of C++ features, this book will help you: * Master the fundamentals of C++, including variables, data types, operators, control flow statements, and functions * Understand object-oriented programming concepts such as classes, inheritance, and polymorphism * Explore advanced C++ techniques such as templates, lambda expressions, and multithreading * Design and implement efficient algorithms, handle errors and exceptions, and optimize your code for performance * Learn about the latest trends and developments in C++ programming, including its applications in artificial intelligence, machine learning, and cloud computing This book is the perfect resource for anyone who wants to master the art of C++ programming. With its comprehensive coverage of C++ concepts and its focus on practical application, this book will help you build the skills you need to succeed in today's competitive software development landscape. Whether you are a student learning C++ for the first time or a seasoned programmer looking to expand your skills, this book is the perfect companion on your journey to C++ mastery. If you like this book, write a review on google books!

The Art of C++: Master the Language Like a Pro

Embark on an extraordinary journey into the realm of C++ programming with this comprehensive guide, crafted to empower you with the knowledge and skills to conquer the complexities of this powerful language. Delve into the fundamentals of C++, mastering data types, structures, and control structures, the building blocks of effective programming. As you progress, unravel the elegance of object-oriented programming, a paradigm that mirrors the real world's complexities, enabling you to structure your code with clarity and precision. Conquer advanced C++ concepts, unlocking the full potential of this versatile language. Grasp the intricacies of templates, enabling you to write generic code that adapts to diverse scenarios. Delve into the depths of memory management, ensuring your programs handle memory resources efficiently and effectively. Explore error handling techniques, safeguarding your applications against unforeseen pitfalls. With each chapter, you'll ascend the ladder of C++ mastery, gaining the skills and knowledge to tackle real-world programming challenges. Build dynamic websites, craft immersive games, delve into the intricacies of operating systems, and harness the power of C++ for scientific computing. Experience the thrill of transforming your ideas into tangible software creations, leaving an indelible mark on the world of technology. Join the ranks of skilled C++ developers, harnessing the language's might to shape the future of software innovation. Whether you're a seasoned programmer seeking to expand your horizons or a novice eager to embark on a journey of discovery, this book will serve as your trusted guide, illuminating the path to C++ mastery. Embrace the challenge, unlock your potential, and become a true C++ virtuoso. In this comprehensive guide, you'll discover: * The fundamentals of C++, including data types, structures, and control structures * The art of object-oriented programming, a paradigm that mirrors the real world's complexities * Advanced C++ concepts, such as templates, memory management, and error handling * Practical applications of C++ in various domains, including web development, game development, and scientific computing * Expert guidance and insights to help you master C++ and become a proficient programmer With clear explanations, step-by-step examples, and hands-on exercises, this book is your ultimate companion on the path to C++ mastery. Start your journey today and unlock the power of this versatile language! If you like this book, write a review!

Refactoring with C++

Improve readability and understandability of code using C++ best practices Key Features Enrich your coding skills using features from the modern C++ standard and industry approved libraries Implement refactoring techniques and SOLID principles in C++ Apply automated tools to improve your code quality Purchase of the print or Kindle book includes a free PDF eBook Book DescriptionDespite the prevalence of higher-level languages, C++ is still running the world, from bare-metal embedded systems to distributed cloud-native systems. C++ is on the frontline whenever there is a need for a performance-sensitive tool supporting complex data structures. The language has been actively evolving for the last two decades. This book is a comprehensive guide that shows you how to implement SOLID principles and refactor legacy code using the modern features and approaches of C++, the standard library, Boost library collection, and Guidelines Support Library by Microsoft. The book begins by describing the essential elements of writing clean code and discussing object-oriented programming in C++. You’ll explore the design principles of software testing with examples of using popular unit testing frameworks such as Google Test. The book also guides you through applying automated tools for static and dynamic code analysis using Clang Tools. By the end of this book, you’ll be proficient in applying industry-approved coding practices to design clean, sustainable, and readable real-world C++ code.What you will learn Leverage the rich type system of C++ to write safe and elegant code Create advanced object-oriented designs using the unique features of C++ Minimize code duplication by using metaprogramming Refactor code safely with the help of unit tests Ensure code conventions and format with clang-format Facilitate the usage of modern features automatically with clang-tidy Catch complex bugs such as memory leakage and data races with Clang AddressSanitizer and ThreadSanitizer Who this book is for This book will benefit experienced C++ programmers the most, but is also suitable for technical leaders, software architects, and senior software engineers who want to save on costs and improve software development process efficiency by using modern C++ features and automated tools.

The Art of Software Testing

The classic, landmark work on software testing The hardware and software of computing have changed markedly in the three decades since the first edition of The Art of Software Testing, but this book's powerful underlying analysis has stood the test of time. Whereas most books on software testing target particular development techniques, languages, or testing methods, The Art of Software Testing, Third Edition provides a brief but powerful and comprehensive presentation of time-proven software testing approaches. If your software development project is mission critical, this book is an investment that will pay for itself with the first bug you find. The new Third Edition explains how to apply the book's classic principles to today's hot topics including: Testing apps for iPhones, iPads, BlackBerrys, Androids, and other mobile devices Collaborative (user) programming and testing Testing for Internet applications, e-commerce, and agile programming environments Whether you're a student looking for a testing guide you'll use for the rest of your career, or an IT manager overseeing a software development team, The Art of Software Testing, Third Edition is an expensive book that will pay for itself many times over.

Introduction to the Art of Programming Using Scala

With its flexibility for programming both small and large projects, Scala is an ideal language for teaching beginning programming. Yet there are no textbooks on Scala currently available for the CS1/CS2 levels. Introduction to the Art of Programming Using Scala presents many concepts from CS1 and CS2 using a modern, JVM-based language that works well for both programming in the small and programming in the large. The book progresses from true programming in the small to more significant projects later, leveraging the full benefits of object orientation. It first focuses on fundamental problem solving and programming in the small using the REPL and scripting environments. It covers basic logic and problem decomposition and explains how to use GUIs and graphics in programs. The text then illustrates the benefits of object-oriented design and presents a large collection of basic data structures showing different implementations of key ADTs along with more atypical data structures. It also introduces multithreading and networking to provide further motivating examples. By using Scala as the language for both CS1 and CS2 topics, this textbook gives students an easy entry into programming small projects as well as a firm foundation for taking on larger-scale projects. Many student and instructor resources are available at www.programmingusingscala.net

C# for Artists

A comprehensive guide with extensive coverage of concepts such as OOP, functional programming, generic programming, concurrency, and STL along with the latest features of C++ Purchase of the print or Kindle book includes a free PDF eBook Key Features Delve into the core patterns and components of C++ to master application design Learn tricks, techniques, and best practices to solve common design and architectural challenges Understand the limitation imposed by C++ and how to solve them using design patterns Book DescriptionC++ is a general-purpose programming language designed for efficiency, performance, and flexibility. Design patterns are commonly accepted solutions to well-recognized design problems. In essence, they are a library of reusable components, only for software architecture, and not for a concrete implementation. This book helps you focus on the design patterns that naturally adapt to your needs, and on the patterns that uniquely benefit from the features of C++. Armed with the knowledge of these patterns, you’ll spend less time searching for solutions to common problems and tackle challenges with the solutions developed from experience. You’ll also explore that design patterns are a concise and efficient way to communicate, as patterns are a familiar and recognizable solution to a specific problem and can convey a considerable amount of information with a single line of code. By the end of this book, you’ll have a deep understanding of how to use design patterns to write maintainable, robust, and reusable software. What you will learn Recognize the most common design patterns used in C++ Understand how to use C++ generic programming to solve common design problems Explore the most powerful C++ idioms, their strengths, and their drawbacks Rediscover how to use popular C++ idioms with generic programming Discover new patterns and idioms made possible by language features of C++17 and C++20 Understand the impact of design patterns on the program's performance Who this book is for This book is for experienced C++ developers and programmers who wish to learn about software design patterns and principles and apply them to create robust, reusable, and easily maintainable programs and software systems.

Hands-On Design Patterns with C++

The first book in the IT Architect series helps aspiring & experienced IT infrastructure architects/administrators, and those pursuing infrastructure design certifications, establish a solid foundation in the art of infrastructure design. The three autho

IT Architect: Foundation in the Art of Infrastructure Design: A Practical Guide for IT Architects

Build fast and effective applications using Masonite, a Python-based framework. This book covers creating a digital home assistant application, but it’s certainly not the only kind of application you could build. By working on this kind of project, you’ll cover the broad range of topics and requirements you’re likely to find as you establish your own web empire. You’ll see how Masonite is a developer-centric Python framework, which provides all the tools you’ll need to build powerful and maintainable web applications. After reading and using this book, you'll have the tools to make and deploy your own web ecommerce application from scratch using the Masonite framework. What You Will Learn Customize your request and response cycle with middleware, route groups, and headers Add security into your application and protect against possible vulnerabilities Create and control your application’s maintenance features from a terminal Manage a database, so your teammates and environments will always be in sync Represent relational database data with object-oriented programming techniques Read and write from the file system Process large datasets and slow operations in the background Add real-time features and recurring tasks to your application Who This Book Is For Web developers new to the Masonite framework. Some prior Python experience is helpful but is not required.

The Definitive Guide to Masonite

Education has a substantial impact and influences on almost all sectors in modern society. Different computer-supported educational systems have been developing for many decades to support and make easier teaching and learning processes on all levels of education. Influences of rapid development of Information Communication Technologies and other related disciplines on design and implementation of intelligent, sophisticated educational systems are evident. Nowadays intensive development and wide applications of Artificial Intelligent techniques significantly affect the development of intelligent tutoring systems, smart learning environments that incorporate virtual and augmented reality and robots. Artificial Intelligence has the potential to address some of the biggest challenges in education today, but also in the future in order to establish innovative teaching and learning practices facilitated by powerful educational datamining and learning analytics. This book presents a collection of 17 chapters that bring interesting aspects of the state-of-the-art of application of intelligent techniques in different educational processes and settings. We believe that the works presented in the book will be of great interest to readers and that will motivate them to try to enhance presented approaches and propose better and more advanced solutions.

Handbook on Intelligent Techniques in the Educational Process

This book constitutes the refereed proceedings of the 25th International Conference on Advanced Information Systems Engineering, CAiSE 2013, held in Valencia, Spain, in June 2013. The 44 revised full papers were carefully reviewed and selected from 162 submissions. The contributions have been grouped into the following topical sections: services; awareness; business process execution; products; business process modelling; modelling languages and meta models; requirements engineering 1; enterprise architecture; information systems evolution; mining and predicting; data warehouses and business intelligence; requirements engineering 2; knowledge and know-how; information systems quality; and human factors.

Advanced Information Systems Engineering

Contains C++ source programs that exercise and demonstrate all of the subroutines, procedures, and functions in Numerical Recipes in C++.

Numerical Recipes Example Book (C++)

Ensure your code is sleek, efficient and elegant by mastering powerful Python design patterns About This Book Learn all about abstract design patterns and how to implement them in Python 3 Understand the structural, creational, and behavioral Python design patterns Get to know the context and application of design patterns to solve real-world problems in software architecture, design, and application development Discover how to simplify Design Pattern implementation using the power of Python 3 Who This Book Is For If you have basic Python skills and wish to learn in depth how to correctly apply appropriate design patterns, this course is tailor made for you. What You Will Learn Discover what design patterns are and how to apply them to writing Python Implement objects in Python by creating classes and defining methods Separate related objects into a taxonomy of classes and describe the properties and behaviors of those objects via the class interface Understand when to use object-oriented features, and more importantly when not to use them Get to know proven solutions to common design issues Explore the design principles that form the basis of software design, such as loose coupling, the Hollywood principle, and the Open Close principle, among others Use Structural Design Patterns and find out how objects and classes interact to build larger applications Improve the productivity and code base of your application using Python design patterns Secure an interface using the Proxy pattern In Detail Python is an object-oriented scripting language that is used in everything from data science to web development. Known for its simplicity, Python increases productivity and minimizes development time. Through applying essential software engineering design patterns to Python, Python code becomes even more efficient and reusable from project to project. This learning path takes you through every traditional and advanced design pattern best applied to Python code, building your skills in writing exceptional Python. Divided into three distinct modules, you'll go from foundational to advanced concepts by following a series of practical tutorials. Start with the bedrock of Python programming – the object-oriented paradigm. Rethink the way you work with Python as you work through the Python data structures and object-oriented techniques essential to modern Python programming. Build your confidence as you learn Python syntax, and how to use OOP principles with Python tools such as Django and Kivy. In the second module, run through the most common and most useful design patterns from a Python perspective. Progress through Singleton patterns, Factory patterns, Facade patterns and more all with detailed hands-on guidance. Enhance your professional abilities in in software architecture, design, and development. In the final module, run through the more complex and less common design patterns, discovering how to apply them to Python coding with the help of real-world examples. Get to grips with the best practices of writing Python, as well as creating systems architecture and troubleshooting issues. This Learning Path combines some of the best that Packt has to offer in one complete, curated package. It includes content from the following Packt products: Python 3 Object-Oriented Programming - Second Edition by Dusty Phillips Learning Python Design Patterns - Second Edition by Chetan Giridhar Mastering Python Design Patterns by Sakis Kasampalis Style and approach Advance your Python code through three distinct modules that each build on preceding content. Get the complete coverage of Python design patterns you need to write elegant and efficient code that's reusable and powerful.

The British National Bibliography

The two-volume set LNCS 15701 + 15702 constitutes the proceedings of the 37th International Conference on Advanced Information Systems Engineering, CAiSE 2025, which was held in Vienna, Austria, during June 16–17, 2025. The 35 papers included in the proceedings were carefully reviewed and selected from 229 submissions. They were organized in topical sections as follows: Part I: Modelling with LLM; Security; Sustainability; Chatbots and social networks; process monitoring; IS-development and usage; pre-processing and forecasting; Part II: Comprehension, explanation and recommendation; process discovery; system architecture and privacy; conformance-checking; cloud systems; extending process modelling; ontologies and knowledge graphs.

Python: Master the Art of Design Patterns

In this practical guide, four Kubernetes professionals with deep experience in distributed systems, enterprise application development, and open source will guide you through the process of building applications with this container orchestration system. They distill decades of experience from companies that are successfully running Kubernetes in production and provide concrete code examples to back the methods presented in this book. Revised to cover all the latest Kubernetes features, new tooling, and deprecations, this book is ideal for those who are familiar with basic Kubernetes concepts but want to get up to speed on the latest best practices. You'll learn exactly what you need to know to build your best app with Kubernetes the first time. Set up and develop applications in Kubernetes Learn patterns for monitoring, securing your systems, and managing upgrades, rollouts, and rollbacks Integrate services and legacy applications and develop higher-level platforms on top of Kubernetes Run machine learning workloads in Kubernetes Ensure pod and container security Understand issues that have become increasingly critical to the successful implementation of Kubernetes, such as chaos engineering/testing, GitOps, service mesh, and observability

Advanced Information Systems Engineering

Unlock the power of Solidity programming and elevate your expertise with \"Mastering the Art of Solidity Programming: Advanced Techniques and Skills.\" Designed for seasoned developers, this book delves deeply into the sophisticated facets of Solidity, offering a comprehensive understanding of the language's advanced syntax, design patterns, and security best practices. It empowers readers to build efficient, scalable, and secure smart contracts that leverage the full potential of the Ethereum blockchain. Embark on a thorough exploration of cutting-edge topics, from optimizing gas usage to integrating blockchain oracles and implementing proxy patterns for contract upgradeability. Each chapter meticulously examines real-world applications within the decentralized finance landscape, providing practical tools and strategies to address current and emerging challenges. With detailed insights into testing, debugging, and managing smart contracts in the Ethereum ecosystem, this guide ensures reliable, robust code development. Whether you're enhancing DeFi solutions, advancing blockchain interoperability, or pioneering new scalability pathways, this book equips you with the knowledge and skills required to innovate confidently in today's dynamic blockchain domain. \"Mastering the Art of Solidity Programming\" stands as an essential resource for developers seeking to push the boundaries of what is possible with Solidity, positioning you at the forefront of blockchain innovation.

Kubernetes Best Practices

For more than 20 years, Network World has been the premier provider of information, intelligence and insight for network and IT executives responsible for the digital nervous systems of large organizations. Readers are responsible for designing, implementing and managing the voice, data and video systems their companies use to support everything from business critical applications to employee collaboration and electronic commerce.

Dr. Dobb's Journal

Mastering the Art of Solidity Programming: Advanced Techniques and Skills
https://www.heritagefarmmuseum.com/!46520450/ischedulev/lcontinuen/tanticipateh/fundamentals+of+corporate+fi
https://www.heritagefarmmuseum.com/~70947593/uregulateh/kfacilitateo/funderlineb/practical+jaguar+ownership+
https://www.heritagefarmmuseum.com/$76362446/dcirculatev/ucontinuek/funderlinee/pedoman+pengobatan+dasar+
https://www.heritagefarmmuseum.com/=63452970/dguaranteeh/rperceivem/vanticipateo/complete+guide+to+credit+
https://www.heritagefarmmuseum.com/_45948399/tconvincer/bfacilitatec/oestimatel/mercedes+300d+owners+manu
https://www.heritagefarmmuseum.com/-21059532/wregulated/qdescribet/ccriticisem/download+2005+kia+spectra+manual.pdf
https://www.heritagefarmmuseum.com/-59466834/cpreservet/shesitatea/kencountery/hyster+forklift+repair+manuals.pdf
https://www.heritagefarmmuseum.com/$71021055/epreserved/jcontinuev/canticipatem/the+elementary+teachers+of+
https://www.heritagefarmmuseum.com/$82179773/eguaranteeb/ocontinuen/pencounterj/manual+sprinter.pdf
https://www.heritagefarmmuseum.com/-

94708427/vguaranteep/wdescribec/ycriticises/microeconomics+pindyck+8th+edition+solutions.pdf