Code Complete A Practical Handbook Of Software Construction

A Step-by-Step Guide to Qualitative Data Coding is a comprehensive qualitative data analysis guide. It is designed to help readers to systematically analyze qualitative data in a transparent and consistent manner, thus promoting the credibility of their findings. The book examines the art of coding data, categorizing codes, and synthesizing categories and themes. Using real data for demonstrations, it provides step-by-step instructions and illustrations for analyzing qualitative data. Some of the demonstrations include conducting manual coding using Microsoft Word and how to use qualitative data analysis software such as Dedoose, NVivo and QDA Miner Lite to analyze data. It also contains creative ways of presenting qualitative findings and provides practical examples. After reading this book, readers will be able to: Analyze qualitative data and present their findings Select an appropriate qualitative analysis tool Decide on the right qualitative coding and categorization strategies for their analysis Develop relationships among categories/themes Choose a suitable format for the presentation of the findings It is a great resource for qualitative research instructors and undergraduate and graduate students who want to gain skills in analyzing qualitative data or who plan to conduct a qualitative study. It is also useful for researchers and practitioners in the social and health sciences fields.

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

These are the proven, effective agile practices that will make you a better developer. You'll learn pragmatic ways of approaching the development process and your personal coding techniques. You'll learn about your own attitudes, issues with working on a team, and how to best manage your learning, all in an iterative, incremental, agile style. You'll see how to apply each practice, and what benefits you can expect. Bottom line: This book will make you a better developer.

Presents practical advice on the disciplines, techniques, tools, and practices of computer programming and how to approach software development with a sense of pride, honor, and self-respect.

Users can dramatically improve the design, performance, and manageability of object-oriented code without altering its interfaces or behavior. "Refactoring" shows users exactly how to spot the best opportunities for refactoring and exactly how to do it, step by step.

The demand for digital speech coding algorithms grows every day, fueled by applications such as streaming speech over the Internet, Internet telephone, digital cellular telephony, wireless teleconferencing, and various multimedia applications. Until now, most of the books available on audio coding have been collections of individually authored pape

An essential guide for teaching and learning computational art and design: exercises, assignments, interviews, and more than 170 illustrations of creative work. This book is an essential resource for art educators and practitioners who want to explore code as a creative medium, and serves as a guide for computer scientists transitioning from STEM to STEAM in their syllabi or practice. It provides a collection of classic creative coding prompts and assignments, accompanied by annotated examples of both classic and contemporary projects, and more than 170 illustrations of creative work, and features a set of interviews with leading educators. Picking up where standard programming guides leave off, the authors highlight alternative programming pedagogies suitable for the art- and design-oriented classroom, including teaching approaches, resources, and community support structures.

The second edition of this best-selling Python book (over 500,000 copies sold!) uses Python 3 to teach even the technically uninclined how to write programs that do in minutes what would take hours to do by hand. There is no prior programming experience required and the book is loved by liberal arts majors and geeks alike. If you've ever spent hours renaming files or updating hundreds of spreadsheet cells, you know how tedious tasks like these can be. But what if you could have your computer do them for you? In this fully revised second edition of the best-selling classic Automate the Boring Stuff with Python, you'll learn how to use Python to write programs that do in minutes what would take you hours to do by hand--no prior programming experience required. You'll learn the basics of Python and explore Python's rich library of modules for performing specific tasks, like scraping data off websites, reading PDF and Word documents, and automating clicking and typing tasks. The second edition of this international fan favorite includes a brand-new chapter on input validation, as well as tutorials on automating Gmail and Google Sheets, plus tips on automatically updating CSV files. You'll learn how to create programs that effortlessly perform useful feats of automation to: • Search for text in a file or across multiple files • Create, update, move, and rename files and folders • Search the Web and download online content • Update and format data in Excel spreadsheets of any size • Split, merge, watermark, and encrypt PDFs • Send email responses and text notifications • Fill out online forms Step-by-step instructions walk you through each program, and updated practice projects at the end of each chapter challenge you to improve those programs and use your newfound skills to automate similar tasks. Don't spend your time doing work a well-trained monkey could do. Even if you've never written a line of code, you can make your computer do the grunt work. Learn how in Automate the Boring Stuff with Python, 2nd Edition.

This simple and essential book about the craft of acting describes a technique developed and refined by the authors, all of them young actors, in their work with Pulitzer Prize-winning playwright David Mamet, actor W. H. Macy, and director Gregory Mosher. A Practical Handbook for the Actor is written for any actor who has ever experienced the frustrations of acting classes that lacked clarity and objectivity, and that failed to provide a dependable set of tools. An actor's job, the authors state, is to "find a way to live truthfully under the imaginary circumstances of the play." The ways in which an actor can attain that truth form the substance of this eloquent book.

A basic typing handbook using the self-teaching, learn-at-your-own-speed methods of one of New York’s most successful business schools. This practical guide offers specialized drills, speed and accuracy timings, centering and tabulating, finished business letters, how to make corrections and copies, proofreaders’ symbols, as well as trouble-saving tips.

Looks at the principles and clean code, includes case studies showcasing the practices of writing clean code, and contains a list of heuristics and "smells" accumulated from the process of writing clean code.

Helps readers develop a solid foundation in programming, teaching concepts that can be used with any modern programming language, covering such topics as text editors, build tools, programming standards, regular expressions, and debugging.

This highly visual handbook integrates cardiac anatomy and the state-of-the-art imaging techniques used in today's catheter or electrophysiology laboratory, guiding readers to a comprehensive understanding of both normal cardiac anatomy and the structures associated with complex heart disease. Well organized, easily navigable, and superbly illustrated in a landscape format, this unique text invites the reader on a visual intracardiac journey via stunning images and schematic illustrations, including such imaging modalities as computed tomography, magnetic resonance imaging, ultrasound, radiogra.

Practical Handbook of Genetic Algorithms, Volume 3: Complex Coding Systems contains computer-code examples for the development of genetic algorithm systems - compiling them from an array of practitioners in the field. Each contribution of this singular resource includes: unique code segments documentation descripti

Corporate and commercial software-development teams all want solutions for one important problem—how to get their high-pressure development schedules under control. In RAPID DEVELOPMENT, author Steve McConnell addresses that concern head-on with overall strategies, specific best practices, and valuable tips that help shrink and control development schedules and keep projects moving. Inside, you'll find: A rapid-development strategy that can be applied to any project and the best practices to make that strategy work Candid discussions of great and not-so-great rapid-development practices—estimation, prototyping, forced overtime, motivation, teamwork, rapid-development languages, risk management, and many others A list of classic mistakes to avoid for rapid-development projects, including creeping requirements, shortchanged quality, and silver-bullet syndrome Case studies that vividly illustrate what can go wrong, what can go right, and how to tell which direction your project is going RAPID DEVELOPMENT is the real-world guide to more efficient applications development.

“One of the most significant books in my life.” –Obie Fernandez, Author, The Rails Way “Twenty years ago, the first edition of The Pragmatic Programmer completely changed the trajectory of my career. This new edition could do the same for yours.” –Mike Cohn, Author of Succeeding with Agile, Agile Estimating and Planning, and User Stories Applied “. . . filled with practical advice, both technical and professional, that will serve you and your projects well for years to come.” –Andrea Goulet, CEO, Corgibytes, Founder, LegacyCode.Rocks “. . . lightning does strike twice, and this book is proof.” –VM (Vicky) Brasseur, Director of Open Source Strategy, Juniper Networks The Pragmatic Programmer is one of those rare tech books you'll read, re-read, and read again over the years. Whether you're new to the field or an experienced practitioner, you'll come away with fresh insights each and every time. Dave Thomas and Andy Hunt wrote the first edition of this influential book in 1999 to help their clients create better software and rediscover the joy of coding. These lessons have helped a generation of programmers examine the very essence of software development, independent of any particular language, framework, or methodology, and the Pragmatic philosophy has spawned hundreds of books, screencasts, and audio books, as well as thousands of careers and success stories. Now, twenty years later, this new edition re-examines what it means to be a modern programmer. Topics range from personal responsibility and career development to architectural techniques for keeping your code flexible and easy to adapt and reuse. Read this book, and you'll learn how to: Fight software rot Learn continuously Avoid the trap of duplicating knowledge Write flexible, dynamic, and adaptable code Harness the power of basic tools Avoid programming by coincidence Learn real requirements Solve the underlying problems of concurrent code Guard against security vulnerabilities Build teams of Pragmatic Programmers Take responsibility for your work and career Test ruthlessly and effectively, including property-based testing Implement the Pragmatic Starter Kit Delight your users Written as a series of self-contained sections and filled with classic and fresh anecdotes, thoughtful examples, and interesting analogies, The Pragmatic Programmer illustrates the best approaches and major pitfalls of many different aspects of software development. Whether you’re a new coder, an experienced programmer, or a manager responsible for software projects, use these lessons daily, and you’ll quickly see improvements in personal productivity, accuracy, and job satisfaction. You’ll learn skills and develop habits and attitudes that form the foundation for long-term success in your career. You’ll become a Pragmatic Programmer. Register your book for convenient access to downloads, updates, and/or corrections as they become available. See inside book for details.

"Two thumbs up" —Gregory V. Wilson, Dr. Dobbs Journal (October 2004) No one can disparage the ability to write good code. At itshighest levels, it is an art. But no one can confuse writing good code with developing goodsoftware. The difference—in terms of challenges, skills, andcompensation—is immense. Coder to Developer helps you excel at the manynon-coding tasks entailed, from start to finish, in just about anysuccessful development project. What's more, it equips you with themindset and self-assurance required to pull it all together, sothat you see every piece of your work as part of a coherentprocess. Inside, you'll find plenty of technical guidance on suchtopics as: Choosing and using a source code control system Code generation tools--when and why Preventing bugs with unit testing Tracking, fixing, and learning from bugs Application activity logging Streamlining and systematizing the build process Traditional installations and alternative approaches To pull all of this together, the author has provided the sourcecode for Download Tracker, a tool for organizing your collection ofdownloaded code, that's used for examples throughout this book. Thecode is provided in various states of completion, reflecting everystage of development, so that you can dig deep into the actualprocess of building software. But you'll also develop "softer"skills, in areas such as team management, open sourcecollaboration, user and developer documentation, and intellectualproperty protection. If you want to become someone who can delivernot just good code but also a good product, this book is the placeto start. If you must build successful software projects, it'sessential reading.

Features the best practices in the art and science of constructing software--topics include design, applying good techniques to construction, eliminating errors, planning, managing construction activities, and relating personal character to superior software. Original. (Intermediate)

Widely considered one of the best practical guides to programming, Steve McConnell’s original CODE COMPLETE has been helping developers write better software for more than a decade. Now this classic book has been fully updated and revised with leading-edge practices—and hundreds of new code samples—illustrating the art and science of software construction. Capturing the body of knowledge available from research, academia, and everyday commercial practice, McConnell synthesizes the most effective techniques and must-know principles into clear, pragmatic guidance. No matter what your experience level, development environment, or project size, this book will inform and stimulate your thinking—and help you build the highest quality code. Discover the timeless techniques and strategies that help you: Design for minimum complexity and maximum creativity Reap the benefits of collaborative development Apply defensive programming techniques to reduce and flush out errors Exploit opportunities to refactor—or evolve—code, and do it safely Use construction practices that are right-weight for your project Debug problems quickly and effectively Resolve critical construction issues early and correctly Build quality into the beginning, middle, and end of your project

Often referred to as the “black art” because of its complexity and uncertainty, software estimation is not as difficult or puzzling as people think. In fact, generating accurate estimates is straightforward—once you understand the art of creating them. In his highly anticipated book, acclaimed author Steve McConnell unravels the mystery to successful software estimation—distilling academic information and real-world experience into a practical guide for working software professionals. Instead of arcane treatises and rigid modeling techniques, this guide highlights a proven set of procedures, understandable formulas, and heuristics that individuals and development teams can apply to their projects to help achieve estimation proficiency. Discover how to: Estimate schedule and cost—or estimate the functionality that can be delivered within a given time frame Avoid common software estimation mistakes Learn estimation techniques for you, your team, and your organization * Estimate specific project activities—including development, management, and defect correction Apply estimation approaches to any type of project—small or large, agile or traditional Navigate the shark-infested political waters that surround project estimates When many corporate software projects are failing, McConnell shows you what works for successful software estimation.

Rapid developments in the field of genetic algorithms along with the popularity of the first edition precipitated this completely revised, thoroughly updated second edition of The Practical Handbook of Genetic Algorithms. Like its predecessor, this edition helps practitioners stay up to date on recent developments in the field and provides material

Good software design is simple and easy to understand. Unfortunately, the average computer program today is so complex that no one could possibly comprehend how all the code works. This concise guide helps you understand the fundamentals of good design through scientific laws—principles you can apply to any programming language or project from here to eternity. Whether you’re a junior programmer, senior software engineer, or non-technical manager, you’ll learn how to create a sound plan for your software project, and make better decisions about the pattern and structure of your system. Discover why good software design has become the missing science Understand the ultimate purpose of software and the goals of good design Determine the value of your design now and in the future Examine real-world examples that demonstrate how a system changes over time Create designs that allow for the most change in the environment with the least change in the software Make easier changes in the future by keeping your code simpler now Gain better knowledge of your software’s behavior with more accurate tests

A book about programming, improving skill, and avoiding mistakes.The author spent two years researching every bug avoidance technique she could find. This book contains the best of them.If you want to program faster, with fewer bugs, and write more secure code, buy this book!http://www.zerobugsandprogramfaster.net

Corporate Governance - A Practical Handbook is a user-friendly resource for those needing a practical set of tools to carry out the complex work of the board of directors. The writing is simple and direct with information icons to indicate particularly important passages. Drawing on research and international best commercial practice, this practical handbook provides clear, pragmatic guidance, effective techniques and must-know principles for good governance. No matter what your experience level — whether in a large corporate or a community not-for-profit - this book will inform and stimulate your thinking and help you build the best governance knowledge and practices for your organisation. Practical checklists, templates and tables enable the reader to develop a comprehensive set of governance tools and documents (eg performing a governance audit, developing business strategy and governance policies, recording minutes).

CRC Practical Handbooks are a series of single-volume bench manuals that feature a synthesis of the most frequently used, basic reference information. These highly abridged versions of existing CRC multi-volume Handbooks contain largely tabular and graphic data. They provide extensive coverage in a scientific discipline and enable quick, convenient access to the most practical reference information...on the spot! Leading professionals in their respective fields collaborated to provide individuals and institutions with an economical and easy-to-use source of classic reference information. The CRC Practical Handbook of PHYSICAL PROPERTIES of ROCKS and MINERALS, prepared by leaders in their specialties, has been constructed to serve as a convenient, compact, yet comprehensive source of basic information. The technical data have been compiled and selectively edited to provide an organized and definitive presentation of the physical properties of rocks and their constituent minerals. The format is primarily tabular and graphical, for easy reference and comparisons. There is also instructive textual material to present, explain, and clarify the data. This edited and abridged version of the CRC Handbook of Physical Properties of Rocks, published in three volumes in 1982 - 1984, will serve as an easy-to-use source of current and useful reference information.

The heavily-revised Practical Handbook of Marine Science, Fourth Edition continues its tradition as a state-of-the-art reference that updates the field of marine science to meet the interdisciplinary research needs of physical oceanographers, marine biologists, marine chemists, and marine geologists. This edition adds an entirely new section devoted to Climate Change and Climate Change Effects. It also adds new sections on Estuaries, Beaches, Barrier Islands, Shellfish, Macroalgae, Food Chains, Food Webs, Trophic Dynamics, System Productivity, Physical-Chemical-Biological Alteration, and Coastal Resource Management. The Handbook assembles an extensive international collection of marine science data throughout, with approximately 1,000 tables and illustrations. It provides comprehensive coverage of anthropogenic impacts in estuarine and marine ecosystems from local, regional, and global perspectives. Maintaining its user-friendly, multi-sectional format, this comprehensive resource will also be of value to undergraduate and graduate students, research scientists, administrators, and other professionals who deal with the management of marine resources. Now published in full color, the new edition offers extensive illustrative and tabular reference material covering all the major disciplines related to the sea.

Software developers need to worry about security as never before. They need clear guidance on safe coding practices, and that’s exactly what this book delivers. The book does not delve deep into theory, or rant about the politics of security. Instead, it clearly and simply lays out the most common threats that programmers need to defend against. It then shows programmers how to make their defense. The book takes a broad focus, ranging over SQL injection, worms and buffer overflows, password security, and more. It sets programmers on the path towards successfully defending against the entire gamut of security threats that they might face.

In this comprehensive yet accessible overview for software leaders, the author presents an impactful, action-oriented prescription-covering the practical considerations needed to ensure you reap the full benefits of effective Agile

A guide to writing computer code covers such topics as variable naming, presentation style, error handling, and security.

This crucial reference focuses on the growth of vegetation on disturbed lands, specifically the problems of plant seeding and growth and the ecological consequences of that growth. This one-of-a-kind book covers the spectrum of plant development, including the creation of an acceptable rooting medium and seeding or planting, and discusses practices to enhance diversity and usefulness of the plant community. This handbook gives detailed examples of how to rehabilitate and restore damaged land and, in most cases, mined land. The book focuses on the complete restoration of soil, water, vegetation, and wildlife. It provides a thorough introduction to the science behind the practice and follows with practical examples. A complete, detailed list of genera and species to expedite restoration is also included.

When programmers list their favorite books, Jon Bentley’s collection of programming pearls is commonly included among the classics. Just as natural pearls grow from grains of sand that irritate oysters, programming pearls have grown from real problems that have irritated real programmers. With origins beyond solid engineering, in the realm of insight and creativity, Bentley’s pearls offer unique and clever solutions to those nagging problems. Illustrated by programs designed as much for fun as for instruction, the book is filled with lucid and witty descriptions of practical programming techniques and fundamental design principles. It is not at all surprising that Programming Pearls has been so highly valued by programmers at every level of experience. In this revision, the first in 14 years, Bentley has substantially updated his essays to reflect current programming methods and environments. In addition, there are three new essays on testing, debugging, and timing set representations string problems All the original programs have been rewritten, and an equal amount of new code has been generated. Implementations of all the programs, in C or C++, are now available on the Web. What remains the same in this new edition is Bentley’s focus on the hard core of programming problems and his delivery of workable solutions to those problems. Whether you are new to Bentley’s classic or are revisiting his work for some fresh insight, the book is sure to make your own list of favorites.

How do the experts solve difficult problems in software development? In this unique and insightful book, leading computer scientists offer case studies that reveal how they found unusual, carefully designed solutions to high-profile projects. You will be able to look over the shoulder of major coding and design experts to see problems through their eyes. This is not simply another design patterns book, or another software engineering treatise on the right and wrong way to do things. The authors think aloud as they work through their project's architecture, the tradeoffs made in its construction, and when it was important to break rules. This book contains 33 chapters contributed by Brian Kernighan, KarlFogel, Jon Bentley, Tim Bray, Elliotte Rusty Harold, Michael Feathers,Alberto Savoia, Charles Petzold, Douglas Crockford, Henry S. Warren,Jr., Ashish Gulhati, Lincoln Stein, Jim Kent, Jack Dongarra and PiotrLuszczek, Adam Kolawa, Greg Kroah-Hartman, Diomidis Spinellis, AndrewKuchling, Travis E. Oliphant, Ronald Mak, Rogerio Atem de Carvalho andRafael Monnerat, Bryan Cantrill, Jeff Dean and Sanjay Ghemawat, SimonPeyton Jones, Kent Dybvig, William Otte and Douglas C. Schmidt, AndrewPatzer, Andreas Zeller, Yukihiro Matsumoto, Arun Mehta, TV Raman,Laura Wingerd and Christopher Seiwald, and Brian Hayes. Beautiful Code is an opportunity for master coders to tell their story. All author royalties will be donated to Amnesty International.

T. S. Eliot’s famous collection of nonsense verse about cats—the inspiration for the Andrew Lloyd Webber musical Cats, now made into a major motion picture. This edition features vibrant illustrations by Axel Scheffler.

HTML and CSS can be a little daunting at first but fear not. This book, based on Shay Howe's popular workshop covers the basics and breaks down the barrier to entry, showing readers how they can start using HTML and CSS through practical techniques today. They'll find accompanying code examples online, while they explore topics such as the different structures of HTML and CSS, and common terms. After establishing a basic understanding of HTML and CSS a deeper dive is taken into the box model and how to work with floats. The book includes an exercise focused on cleaning up a web page by improving the user interface and design, solely using HTML and CSS. With a few quick changes the web page changes shape and comes to life. Interactive, technically up-to-the-minute and easy-to-understand, this book will advance a student's skills to a professional level.

Learn how to program by diving into the R language, and then use your newfound skills to solve practical data science problems. With this book, you’ll learn how to load data, assemble and disassemble data objects, navigate R’s environment system, write your own functions, and use all of R’s programming tools. RStudio Master Instructor Garrett Grolemund not only teaches you how to program, but also shows you how to get more from R than just visualizing and modeling data. You’ll gain valuable programming skills and support your work as a data scientist at the same time. Work hands-on with three practical data analysis projects based on casino games Store, retrieve, and change data values in your computer’s memory Write programs and simulations that outperform those written by typical R users Use R programming tools such as if else statements, for loops, and S3 classes Learn how to write lightning-fast vectorized R code Take advantage of R’s package system and debugging tools Practice and apply R programming concepts as you learn them

Widely considered one of the best practical guides to programming, Steve McConnell s original CODE COMPLETE has been helping developers write better software for more than a decade. Now this classic book has been fully updated and revised with leading-edge practices-and hundreds of new code samples-illustrating the art and science of software construction. Capturing the body of knowledge available from research, academia, and everyday commercial practice, McConnell synthesizes the most effective techniques and must-know principles into clear, pragmatic guidance. No matter what your experience level, development environment, or project size, this book will inform and stimulate your thinking-and help you build the highest quality code.

The first complete handbook for every aspect of grouting technology The Practical Handbook of Grouting offers the most comprehensive, single-source reference covering all facets of grouting technology, including its application for control of water movement, strengthening of both soil and rock, and a wide range of structural applications. Richly illustrated with hundreds of informative photographs, graphs, and figures, this handbook provides invaluable advice on all stages of a project from initial investigation and design, through execution, monitoring, and quality control. Broad coverage in the Practical Handbook of Grouting begins with a general overview of the topic and includes design and quality control issues, injection techniques, and a thorough discussion of drilling and grouting equipment, with practical focus on building custom equipment. Enriched with real-world insights from the author, the Practical Handbook of Grouting features the latest information on: * Cementitious and noncementitious grouts, including new admixtures and polymers * Special construction requirements, including grouting inside structures, underground spaces, in extreme environments, and for emergency response support * Grouting equipment, including pumps, mixers, agitators, and delivery and monitoring systems * Pump mechanics, including the advantages and limitations of all pump types * "The Games Contractors Play," including marketing efforts, proposal trickery, on-the-job issues, and defending bad work Complete with an extensive bibliography and references, the Practical Handbook of Grouting is a valuable resource for civil, structural, and geotechnical engineers, geologists, contractors, and students in related fields.

Write code that can adapt to changes. By applying this book’s principles, you can create code that accommodates new requirements and unforeseen scenarios without significant rewrites. Gary McLean Hall describes Agile best practices, principles, and patterns for designing and writing code that can evolve more quickly and easily, with fewer errors, because it doesn’t impede change. Now revised, updated, and expanded, Adaptive Code, Second Edition adds indispensable practical insights on Kanban, dependency inversion, and creating reusable abstractions. Drawing on over a decade of Agile consulting and development experience, McLean Hall has updated his best-seller with deeper coverage of unit testing, refactoring, pure dependency injection, and more. Master powerful new ways to: • Write code that enables and complements Scrum, Kanban, or any other Agile framework • Develop code that can survive major changes in requirements • Plan for adaptability by using dependencies, layering, interfaces, and design patterns • Perform unit testing and refactoring in tandem, gaining more value from both • Use the “golden master” technique to make legacy code adaptive • Build SOLID code with single-responsibility, open/closed, and Liskov substitution principles • Create smaller interfaces to support more-diverse client and architectural needs • Leverage dependency injection best practices to improve code adaptability • Apply dependency inversion with the Stairway pattern, and avoid related anti-patterns About You This book is for programmers of all skill levels seeking more-practical insight into design patterns, SOLID principles, unit testing, refactoring, and related topics. Most readers will have programmed in C#, Java, C++, or similar object-oriented languages, and will be familiar with core procedural programming techniques.