

Coders At Work Reflections On The Craft Of Programming

When people should go to the books stores, search initiation by shop, shelf by shelf, it is in fact problematic. This is why we give the book compilations in this website. It will certainly ease you to look guide **Coders At Work Reflections On The Craft Of Programming** as you such as.

By searching the title, publisher, or authors of guide you really want, you can discover them rapidly. In the house, workplace, or perhaps in your method can be all best area within net connections. If you objective to download and install the Coders At Work Reflections On The Craft Of Programming , it is unconditionally easy then, previously currently we extend the join to buy and make bargains to download and install Coders At Work Reflections On The Craft Of Programming consequently simple!

[The Pragmatic Programmer](#) - David Thomas 2019-07-30

"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.

[Coding Freedom](#) - E. Gabriella Coleman 2013

Who are computer hackers? What is free software? And what does the emergence of a community dedicated to the production of free and open source software--and to hacking as a technical, aesthetic, and moral project--reveal about the values of contemporary liberalism? Exploring the rise and political significance of the free and open source software (F/OSS) movement in the United States and Europe, Coding Freedom details the ethics behind hackers' devotion to F/OSS, the social codes that guide its production, and the political struggles through which hackers question the scope and direction of copyright and patent law. In telling the story of the F/OSS movement, the book unfolds a broader narrative involving computing, the politics of access, and intellectual property. E. Gabriella Coleman tracks the ways in which hackers collaborate and examines passionate manifestos, hacker humor, free software project governance, and festive hacker conferences. Looking at the ways that hackers sustain their productive freedom, Coleman shows that these activists, driven by a commitment to their work, reformulate key ideals including free speech, transparency, and meritocracy, and

refuse restrictive intellectual protections. Coleman demonstrates how hacking, so often marginalized or misunderstood, sheds light on the continuing relevance of liberalism in online collaboration.

[Pragmatic Thinking and Learning](#) - Andy Hunt 2008-10-28

Printed in full color. Software development happens in your head. Not in an editor, IDE, or design tool. You're well educated on how to work with software and hardware, but what about wetware--our own brains? Learning new skills and new technology is critical to your career, and it's all in your head. In this book by Andy Hunt, you'll learn how our brains are wired, and how to take advantage of your brain's architecture. You'll learn new tricks and tipsto learn more, faster, and retain more of what you learn. You need a pragmatic approach to thinking and learning. You need to Refactor Your Wetware. Programmers have to learn constantly; not just the stereotypical new technologies, but also the problem domain of the application, the whims of the user community, the quirks of your teammates, the shifting sands of the industry, and the evolving characteristics of the project itself as it is built. We'll journey together through bits of cognitive and neuroscience, learning and behavioral theory. You'll see some surprising aspects of how our brains work, and how you can take advantage of the system to improve your own learning and thinking skills. In this book you'll learn how to: Use the Dreyfus Model of Skill Acquisition to become more expert Leverage the architecture of the brain to strengthen different thinking modes Avoid common "known bugs" in your mind Learn more deliberately and more effectively Manage knowledge more efficiently

[Managing the Unmanageable](#) - Mickey W. Mantle 2012-09-16

"Mantle and Lichty have assembled a guide that will help you hire, motivate, and mentor a software development team that functions at the highest level. Their rules of thumb and coaching advice are great blueprints for new and experienced software engineering managers alike." —Tom Conrad, CTO, Pandora "I wish I'd had this material available years ago. I see lots and lots of 'meat' in here that I'll use over and over again as I try to become a better manager. The writing style is right on, and I love the personal anecdotes." —Steve Johnson, VP, Custom Solutions, DigitalFish All too often, software development is deemed unmanageable. The news is filled with stories of projects that have run catastrophically over schedule and budget. Although adding some formal discipline to the development process has improved the situation, it has by no means solved the problem. How can it be, with so much time and money spent to get software development under control, that it remains so unmanageable? In *Managing the Unmanageable: Rules, Tools, and Insights for Managing Software People and Teams*, Mickey W. Mantle and Ron Lichty answer that persistent question with a simple observation: You first must make programmers and software teams manageable. That is, you need to begin by understanding your people—how to hire them, motivate them, and lead them to develop and deliver great products. Drawing on their combined seventy years of software development and management experience, and highlighting the insights and wisdom of other successful managers, Mantle and Lichty provide the guidance you need to manage people and teams in order to deliver software successfully. Whether you are new to software management, or have already been working in that role, you will appreciate the real-world knowledge and practical tools packed into this guide.

[Dreaming in Code](#) - Scott Rosenberg 2008-02-26

Our civilization runs on software. Yet the art of creating it continues to be a dark mystery, even to the experts. To find out why it's so hard to bend computers to our will, Scott Rosenberg spent three years following a team of maverick software developers—led by Lotus 1-2-3 creator Mitch Kapor—designing a novel personal information manager meant to challenge market leader Microsoft Outlook. Their story takes us through a maze of abrupt dead ends and exhilarating breakthroughs as they

wrestle not only with the abstraction of code, but with the unpredictability of human behavior— especially their own.

User Interface Design for Mere Mortals - Eric Butow 2007-05-09

User Interface Design for Mere Mortals takes the mystery out of designing effective interfaces for both desktop and web applications. It is recommended reading for anyone who wants to provide users of their software with interfaces that are intuitive and easy-to-use. The key to any successful application lies in providing an interface users not only enjoy interacting with but which also saves time, eliminates frustration, and gets the job done with a minimum of effort. Readers will discover the secrets of good interface design by learning how users behave and the expectations that users have of different types of interfaces. Anyone who reads User Interface Design for Mere Mortals will benefit from

- Gaining an appreciation of the differences in the “look and feel” of interfaces for a variety of systems and platforms
- Learning how to go about designing and creating the most appropriate interface for the application or website being developed
- Becoming familiar with all the different components that make up an interface and the important role that each of those components plays in communicating with users
- Understanding the business benefits that flow from good interface design such as significantly reduced support costs
- Gaining invaluable insights into how users behave, including the seven stages of human interaction with computers
- Working through case study based, in-depth analysis of each of the stages involved in designing a user interface
- Acquiring practical knowledge about the similarities and differences between designing websites and traditional desktop applications
- Learning how to define, conduct, and analyze usability testing

Through the use of the proven For Mere Mortals format, User Interface Design for Mere Mortals succeeds in parting the veil of mystery surrounding effective user interface design. Whatever your background, the For Mere Mortals format makes the information easily accessible and usable. Contents Preface Introduction CHAPTER 1 Brief Histories CHAPTER 2 Concepts and Issues CHAPTER 3 Making the Business Case CHAPTER 4 Good Design CHAPTER 5 How User Behave CHAPTER 6 Analyzing Your Users CHAPTER 7 Designing a User Interface CHAPTER 8 Designing a Web Site CHAPTER 9 Usability APPENDIX A Answers to Review Questions APPENDIX B Recommended Reading Glossary References Index

Coders - Clive Thompson 2020-03-05

Masterful . . . [Thompson] illuminates both the fascinating coders and the bewildering technological forces that are transforming the world in which we live.' David Grann, author of The Lost City of ZFacebook's algorithms shaping the news. Uber's cars flocking the streets. Revolution on Twitter and romance on Tinder. We live in a world constructed of computer code. Coders - software programmers - are the people who built it for us. And yet their worlds and minds are little known to outsiders. In Coders, Wired columnist Clive Thompson presents a brilliantly original anthropological reckoning with the most influential tribe in today's world, interrogating who they are, how they think, what they value, what qualifies as greatness in their world, and what should give us pause. One of the most prominent journalists writing on technology today, Clive Thompson takes us into the minds of coders, the most quietly influential people on the planet, in a journey into the heart of the machine - and the men and women who made it.

The Self-Taught Computer Scientist - Cory Althoff 2021-09-16

The Self-Taught Computer Scientist is Cory Althoff's follow-up to The Self-Taught Programmer, which inspired hundreds of thousands of professionals to learn how to program outside of school. In The Self-Taught Programmer, Cory showed readers why you don't need a computer science degree to program professionally and taught the programming fundamentals he used to go from a complete beginner to a software engineer at eBay without one. In The Self-Taught Computer Scientist, Cory teaches you the computer science concepts that all self-taught programmers should understand to have outstanding careers. The Self-Taught Computer Scientist will not only make you a better programmer; it will also help you pass your technical interview: the interview all programmers have to pass to land a new job. Whether you are preparing to apply for jobs or sharpen your computer science knowledge, reading The Self-Taught Computer Scientist will improve your programming career. It's written for complete beginners, so you should have no problem reading it even if you've never studied computer science before.

Programming Interviews Exposed - John Mongan 2018-03-28

Ace technical interviews with smart preparation Programming Interviews Exposed is the programmer's ideal first choice for technical interview preparation. Updated to reflect changing techniques and trends, this new

fourth edition provides insider guidance on the unique interview process that today's programmers face. Online coding contests are being used to screen candidate pools of thousands, take-home projects have become commonplace, and employers are even evaluating a candidate's public code repositories at GitHub—and with competition becoming increasingly fierce, programmers need to shape themselves into the ideal candidate well in advance of the interview. This book doesn't just give you a collection of questions and answers, it walks you through the process of coming up with the solution so you learn the skills and techniques to shine on whatever problems you're given. This edition combines a thoroughly revised basis in classic questions involving fundamental data structures and algorithms with problems and step-by-step procedures for new topics including probability, data science, statistics, and machine learning which will help you fully prepare for whatever comes your way. Learn what the interviewer needs to hear to move you forward in the process Adopt an effective approach to phone screens with non-technical recruiters Examine common interview problems and tests with expert explanations Be ready to demonstrate your skills verbally, in contests, on GitHub, and more Technical jobs require the skillset, but you won't get hired unless you are able to effectively and efficiently demonstrate that skillset under pressure, in competition with hundreds of others with the same background.

Programming Interviews Exposed teaches you the interview skills you need to stand out as the best applicant to help you get the job you want.

Masterminds of Programming - Federico Biancuzzi 2009-03-21

Masterminds of Programming features exclusive interviews with the creators of several historic and highly influential programming languages. In this unique collection, you'll learn about the processes that led to specific design decisions, including the goals they had in mind, the trade-offs they had to make, and how their experiences have left an impact on programming today. Masterminds of Programming includes individual interviews with: Adin D. Falkoff: APL Thomas E. Kurtz: BASIC Charles H. Moore: FORTH Robin Milner: ML Donald D. Chamberlin: SQL Alfred Aho, Peter Weinberger, and Brian Kernighan: AWK Charles Geschke and John Warnock: PostScript Bjarne Stroustrup: C++ Bertrand Meyer: Eiffel Brad Cox and Tom Love: Objective-C Larry Wall: Perl Simon Peyton Jones, Paul Hudak, Philip Wadler, and John Hughes: Haskell Guido van Rossum: Python Luiz Henrique de Figueiredo and Roberto Ierusalimsky: Lua James Gosling: Java Grady Booch, Ivar Jacobson, and James Rumbaugh: UML Anders Hejlsberg: Delphi inventor and lead developer of C# If you're interested in the people whose vision and hard work helped shape the computer industry, you'll find Masterminds of Programming fascinating.

Lisp in Small Pieces - Christian Queinnec 2003-12-04

This is a comprehensive account of the semantics and the implementation of the whole Lisp family of languages, namely Lisp, Scheme and related dialects. It describes 11 interpreters and 2 compilers, including very recent techniques of interpretation and compilation. The book is in two parts. The first starts from a simple evaluation function and enriches it with multiple name spaces, continuations and side-effects with commented variants, while at the same time the language used to define these features is reduced to a simple lambda-calculus. Denotational semantics is then naturally introduced. The second part focuses more on implementation techniques and discusses precompilation for fast interpretation: threaded code or bytecode; compilation towards C. Some extensions are also described such as dynamic evaluation, reflection, macros and objects. This will become the new standard reference for people wanting to know more about the Lisp family of languages: how they work, how they are implemented, what their variants are and why such variants exist. The full code is supplied (and also available over the Net). A large bibliography is given as well as a considerable number of exercises. Thus it may also be used by students to accompany second courses on Lisp or Scheme.

Beautiful Code - Greg Wilson 2007-06-26

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,

Karl Fogel, Jon Bentley, Tim Bray, Elliott Rusty Harold, Michael Feathers, Alberto Savoia, Charles Petzold, Douglas Crockford, Henry S. Warren, Jr., Ashish Gulhati, Lincoln Stein, Jim Kent, Jack Dongarra and Piotr Luszczek, Adam Kolawa, Greg Kroah-Hartman, Diomidis Spinellis, Andrew Kuchling, Travis E. Oliphant, Ronald Mak, Rogerio Atem de Carvalho and Rafael Monnerat, Bryan Cantrill, Jeff Dean and Sanjay Ghemawat, Simon Peyton Jones, Kent Dybvig, William Otte and Douglas C. Schmidt, Andrew Patzer, 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.

xUnit Test Patterns - Gerard Meszaros 2007-05-21

Automated testing is a cornerstone of agile development. An effective testing strategy will deliver new functionality more aggressively, accelerate user feedback, and improve quality. However, for many developers, creating effective automated tests is a unique and unfamiliar challenge. xUnit Test Patterns is the definitive guide to writing automated tests using xUnit, the most popular unit testing framework in use today. Agile coach and test automation expert Gerard Meszaros describes 68 proven patterns for making tests easier to write, understand, and maintain. He then shows you how to make them more robust and repeatable--and far more cost-effective. Loaded with information, this book feels like three books in one. The first part is a detailed tutorial on test automation that covers everything from test strategy to in-depth test coding. The second part, a catalog of 18 frequently encountered "test smells," provides trouble-shooting guidelines to help you determine the root cause of problems and the most applicable patterns. The third part contains detailed descriptions of each pattern, including refactoring instructions illustrated by extensive code samples in multiple programming languages.

Programmers at Work - Susan M. Lammers 2006

The Coding Manual for Qualitative Researchers - Johnny Saldana 2012-10-04

The Second Edition of Johnny Saldaña's international bestseller provides an in-depth guide to the multiple approaches available for coding qualitative data. Fully up to date, it includes new chapters, more coding techniques and an additional glossary. Clear, practical and authoritative, the book: -describes how coding initiates qualitative data analysis -demonstrates the writing of analytic memos -discusses available analytic software -suggests how best to use The Coding Manual for Qualitative Researchers for particular studies. In total, 32 coding methods are profiled that can be applied to a range of research genres from grounded theory to phenomenology to narrative inquiry. For each approach, Saldaña discusses the method's origins, a description of the method, practical applications, and a clearly illustrated example with analytic follow-up. A unique and invaluable reference for students, teachers, and practitioners of qualitative inquiry, this book is essential reading across the social sciences.

Reflections on the Work of C.A.R. Hoare - Cliff B. Jones 2010-09-28

Written in honor of Sir Tony Hoare's 75th Birthday, this book provides a discussion of the influence of Hoare's work on current research from an international selection of expert contributors. Includes a scientific biography, listing his most influential work.

Coders at Work - Peter Seibel 2010-05-04

Peter Seibel interviews 15 of the most interesting computer programmers alive today in Coders at Work, offering a companion volume to Apress's highly acclaimed best-seller Founders at Work by Jessica Livingston. As the words "at work" suggest, Peter Seibel focuses on how his interviewees tackle the day-to-day work of programming, while revealing much more, like how they became great programmers, how they recognize programming talent in others, and what kinds of problems they find most interesting. Hundreds of people have suggested names of programmers to interview on the Coders at Work web site: www.codersatwork.com. The complete list was 284 names. Having digested everyone's feedback, we selected 15 folks who've been kind enough to agree to be interviewed: Frances Allen: Pioneer in optimizing compilers, first woman to win the Turing Award (2006) and first female IBM fellow Joe Armstrong: Inventor of Erlang Joshua Bloch: Author of the Java collections framework, now at Google Bernie Cosell: One of the main software guys behind the original ARPANET IMPs and a master debugger Douglas Crockford: JSON founder, JavaScript architect at Yahoo! L. Peter Deutsch: Author of Ghostscript, implementer of Smalltalk-80 at Xerox PARC and Lisp 1.5 on PDP-1 Brendan Eich: Inventor of JavaScript, CTO of the Mozilla Corporation Brad Fitzpatrick:

Writer of LiveJournal, OpenID, memcached, and Perlbal Dan Ingalls: Smalltalk implementor and designer Simon Peyton Jones: Coinventor of Haskell and lead designer of Glasgow Haskell Compiler Donald Knuth: Author of The Art of Computer Programming and creator of TeX Peter Norvig: Director of Research at Google and author of the standard text on AI Guy Steele: Coinventor of Scheme and part of the Common Lisp Gang of Five, currently working on Fortress Ken Thompson: Inventor of UNIX Jamie Zawinski: Author of XEmacs and early Netscape/Mozilla hacker

Reflections - Benjamin Bergery 2002

Case studies of film lighting by some of the world's leading cinematographers addressing key topics such as: basic techniques, light sources, portraits, day and night interiors, postproduction techniques and the design of sequences.

97 Things Every Programmer Should Know - Kevlin Henney 2010-02-05

Tap into the wisdom of experts to learn what every programmer should know, no matter what language you use. With the 97 short and extremely useful tips for programmers in this book, you'll expand your skills by adopting new approaches to old problems, learning appropriate best practices, and honing your craft through sound advice. With contributions from some of the most experienced and respected practitioners in the industry--including Michael Feathers, Pete Goodliffe, Diomidis Spinellis, Cay Horstmann, Verity Stob, and many more--this book contains practical knowledge and principles that you can apply to all kinds of projects. A few of the 97 things you should know: "Code in the Language of the Domain" by Dan North "Write Tests for People" by Gerard Meszaros "Convenience Is Not an -ility" by Gregor Hohpe "Know Your IDE" by Heinz Kabutz "A Message to the Future" by Linda Rising "The Boy Scout Rule" by Robert C. Martin (Uncle Bob) "Beware the Share" by Udi Dahan

Rebel Code - Glyn Moody 2009-02-18

The open source saga has many fascinating chapters. It is partly the story of Linus Torvalds, the master hacker who would become chief architect of the Linux operating system. It is also the story of thousands of devoted programmers around the world who spontaneously worked in tandem to complete the race to shape Linux into the ultimate killer app. Rebel Code traces the remarkable roots of this unplanned revolution. It echoes the twists and turns of Linux's improbable development, as it grew through an almost biological process of accretion and finally took its place at the heart of a jigsaw puzzle that would become the centerpiece of open source. With unprecedented access to the principal players, Moody has written a powerful tale of individual innovation versus big business. Rebel Code provides a from-the-trenches perspective and looks ahead to how open source is challenging long-held conceptions of technology, commerce, and culture.

Patterns of Software - Richard P. Gabriel 1998-04-01

In a book that will intrigue anyone who is curious about Silicon Valley, computer programming, or the world of high technology, respected software pioneer and computer scientist Richard Gabriel offers an informative insider's look at the world of software design and computer programming and the business that surrounds them. 10 illustrations.

Seriously Good Software - Marco Faella 2020-03-24

Summary Serious developers know that code can always be improved. With each iteration, you make optimizations—small and large—that can have a huge impact on your application's speed, size, resilience, and maintainability. In Seriously Good Software: Code that Works, Survives, and Wins, author, teacher, and Java expert Marco Faella teaches you techniques for writing better code. You'll start with a simple application and follow it through seven careful refactorings, each designed to explore another dimension of quality. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the technology Great code blends the skill of a programmer with the time-tested techniques and best practices embraced by the entire development community. Although each application has its own context and character, some dimensions of quality are always important. This book concentrates on eight pillars of seriously good software: speed, memory usage, reliability, readability, thread safety, generality, and elegance. The Java-based examples demonstrate techniques that apply to any OO language. About the book Seriously Good Software is a handbook for any professional developer serious about improving application quality. It explores fundamental dimensions of code quality by enhancing a simple implementation into a robust, professional-quality application. Questions, exercises, and Java-based examples ensure you'll get a firm grasp of the concepts as you go.

When you finish the last version of the book's central project, you'll be able to confidently choose the right optimizations for your code. What's inside Evaluating software qualities Assessing trade-offs and interactions Fulfilling different objectives in a single task Java-based exercises you can apply in any OO language About the reader For web developers comfortable with JavaScript and HTML. About the author Marco Faella teaches advanced programming at a major Italian university. His published work includes peer-reviewed research articles, a Java certification manual, and a video course. Table of Contents *Part 1: Preliminaries * 1 Software qualities and a problem to solve 2 Reference implementation *Part 2: Software Qualities* 3 Need for speed: Time efficiency 4 Precious memory: Space efficiency 5 Self-conscious code: Reliability through monitoring 6 Lie to me: Reliability through testing 7 Coding aloud: Readability 8 Many cooks in the kitchen: Thread safety 9 Please recycle: Reusability

The Planet Remade - Oliver Morton 2017-05-02

First published in Great Britain by Granta Books, 2015.

Qualitative Data Analysis with NVivo - Patricia Bazeley 2013-04-30

Lecturers/instructors only - request a free digital inspection copy here

This straightforward, jargon-free book provides an invaluable introduction to planning and conducting qualitative data analysis with NVivo. Written by leading authorities, with over 40 years combined experience in computer-assisted analysis of qualitative and mixed-mode data, the new edition of this best selling textbook is an ideal mix of practical instruction, methodology and real world examples. Practical, clear and focused the book effectively shows how NVivo software can accommodate and assist analysis across a wide range of research questions, data types, perspectives and methodologies. It sets out: The power and flexibility of the NVivo software How best to use NVivo at each stage in your research project Examples from the authors' own research and the sample data that accompanies the software, supplemented with vignettes drawn from across the social sciences Annotated screen shots A website with links to data, sample projects, supplementary/updated instructions, and SAGE journal content This second edition contains new chapters on handling a literature review, visualizing data, working in mixed methods and social media datasets, and approaching NVivo as a team. An insightful step-by-step guide to the messy reality of doing computer-assisted analysis, this successful book is essential reading for anyone considering using NVivo software.

Computer Science Distilled - Wladston Ferreira Filho 2017-01-17

A foolproof walkthrough of must-know computer science concepts. A fast guide for those who don't need the academic formality, it goes straight to what differentiates pros from amateurs. First introducing discrete mathematics, then exposing the most common algorithm and data structure design elements, and finally the working principles of computers and programming languages, the book is indicated to all programmers.

The Passionate Programmer - Chad Fowler 2009-05-28

Success in today's IT environment requires you to view your career as a business endeavor. In this book, you'll learn how to become an entrepreneur, driving your career in the direction of your choosing. You'll learn how to build your software development career step by step, following the same path that you would follow if you were building, marketing, and selling a product. After all, your skills themselves are a product. The choices you make about which technologies to focus on and which business domains to master have at least as much impact on your success as your technical knowledge itself--don't let those choices be accidental. We'll walk through all aspects of the decision-making process, so you can ensure that you're investing your time and energy in the right areas. You'll develop a structured plan for keeping your mind engaged and your skills fresh. You'll learn how to assess your skills in terms of where they fit on the value chain, driving you away from commodity skills and toward those that are in high demand. Through a mix of high-level, thought-provoking essays and tactical "Act on It" sections, you will come away with concrete plans you can put into action immediately. You'll also get a chance to read the perspectives of several highly successful members of our industry from a variety of career paths. As with any product or service, if nobody knows what you're selling, nobody will buy. We'll walk through the often-neglected world of marketing, and you'll create a plan to market yourself both inside your company and to the industry in general. Above all, you'll see how you can set the direction of your career, leading to a more fulfilling and remarkable professional life.

Girls Who Code - Reshma Saujani 2017-08-22

NEW YORK TIMES BESTSELLER! Part how-to, part girl-empowerment,

and all fun, from the leader of the movement championed by Sheryl Sandberg, Malala Yousafzai, and John Legend. Since 2012, the organization Girls Who Code has taught computing skills to and inspired over 40,000 girls across America. Now its founder, and author *Brave Not Perfect*, Reshma Saujani, wants to inspire you to be a girl who codes! Bursting with dynamic artwork, down-to-earth explanations of coding principles, and real-life stories of girls and women working at places like Pixar and NASA, this graphically animated book shows what a huge role computer science plays in our lives and how much fun it can be. No matter your interest—sports, the arts, baking, student government, social justice—coding can help you do what you love and make your dreams come true. Whether you're a girl who's never coded before, a girl who codes, or a parent raising one, this entertaining book, printed in bold two-color and featuring art on every page, will have you itching to create your own apps, games, and robots to make the world a better place.

Algorithms in Java, Parts 1-4 - Robert Sedgewick 2002-07-23

This edition of Robert Sedgewick's popular work provides current and comprehensive coverage of important algorithms for Java programmers. Michael Schidlow and Sedgewick have developed new Java implementations that both express the methods in a concise and direct manner and provide programmers with the practical means to test them on real applications. Many new algorithms are presented, and the explanations of each algorithm are much more detailed than in previous editions. A new text design and detailed, innovative figures, with accompanying commentary, greatly enhance the presentation. The third edition retains the successful blend of theory and practice that has made Sedgewick's work an invaluable resource for more than 400,000 programmers! This particular book, Parts 1-4, represents the essential first half of Sedgewick's complete work. It provides extensive coverage of fundamental data structures and algorithms for sorting, searching, and related applications. Although the substance of the book applies to programming in any language, the implementations by Schidlow and Sedgewick also exploit the natural match between Java classes and abstract data type (ADT) implementations. Highlights Java class implementations of more than 100 important practical algorithms Emphasis on ADTs, modular programming, and object-oriented programming Extensive coverage of arrays, linked lists, trees, and other fundamental data structures Thorough treatment of algorithms for sorting, selection, priority queue ADT implementations, and symbol table ADT implementations (search algorithms) Complete implementations for binomial queues, multiway radix sorting, randomized BSTs, splay trees, skip lists, multiway tries, B trees, extendible hashing, and many other advanced methods Quantitative information about the algorithms that gives you a basis for comparing them More than 1,000 exercises and more than 250 detailed figures to help you learn properties of the algorithms Whether you are learning the algorithms for the first time or wish to have up-to-date reference material that incorporates new programming styles with classic and new algorithms, you will find a wealth of useful information in this book.

Coders at Work - Peter Seibel 2009-09-16

Peter Seibel interviews 15 of the most interesting computer programmers alive today in *Coders at Work*, offering a companion volume to Apress's highly acclaimed best-seller *Founders at Work* by Jessica Livingston. As the words "at work" suggest, Peter Seibel focuses on how his interviewees tackle the day-to-day work of programming, while revealing much more, like how they became great programmers, how they recognize programming talent in others, and what kinds of problems they find most interesting. Hundreds of people have suggested names of programmers to interview on the *Coders at Work* web site: www.codersatwork.com. The complete list was 284 names. Having digested everyone's feedback, we selected 15 folks who've been kind enough to agree to be interviewed: Frances Allen: Pioneer in optimizing compilers, first woman to win the Turing Award (2006) and first female IBM fellow Joe Armstrong: Inventor of Erlang Joshua Bloch: Author of the Java collections framework, now at Google Bernie Cosell: One of the main software guys behind the original ARPANET IMPs and a master debugger Douglas Crockford: JSON founder, JavaScript architect at Yahoo! L. Peter Deutsch: Author of Ghostscript, implementer of Smalltalk-80 at Xerox PARC and Lisp 1.5 on PDP-1 Brendan Eich: Inventor of JavaScript, CTO of the Mozilla Corporation Brad Fitzpatrick: Writer of LiveJournal, OpenID, memcached, and Perlbal Dan Ingalls: Smalltalk implementor and designer Simon Peyton Jones: Coinventor of Haskell and lead designer of Glasgow Haskell Compiler Donald Knuth: Author of *The Art of Computer Programming* and creator of TeX Peter

Norvig: Director of Research at Google and author of the standard text on AI Guy Steele: Coinventor of Scheme and part of the Common Lisp Gang of Five, currently working on Fortress Ken Thompson: Inventor of UNIX Jamie Zawinski: Author of XEmacs and early Netscape/Mozilla hacker

The Secrets of People Who Never Get Sick - Gene Stone 2012-01-15
Who does not want to be healthier? Now in paperback: the book that Andrew Weil calls “offbeat, informative, and fun . . . a great read,” and that has been praised as “a delightful dance through science” (New York Times bestselling author Mark Hyman, M.D.) and as a “remarkable and insightful book [that] offers you the chance to achieve the best health of your life” (Mark Liponis, M.D., Medical Director, Canyon Ranch). Written by bestselling author Gene Stone, *The Secrets of People Who Never Get Sick* arose from his desire to discover what might actually prevent him from getting sick himself. This book, the result of that exploration, tells the stories of twenty-five people who each possess a different secret of excellent health—a secret that makes sense and that Stone discovered has a true scientific underpinning. There are food secrets—why to take garlic and vitamin C, eat more probiotics, become a vegan, drink a tonic of brewer’s yeast. Exercise secrets—the benefits of lifting weights, the power of stretching. Environmental secrets—living in a Blue Zone, understanding the value of germs. Emotional secrets—seek out and stay in touch with friends, cultivate your spirituality. Physical secrets—nap more, take cold showers in the morning. And the wisdom that goes back generations: Yes, chicken soup works. The stories make it personal, the research makes it real, and the do-it-yourself information shows how to integrate each secret into your own life, and become the next person who never gets sick.

Domain-Driven Design Distilled - Vaughn Vernon 2016-06-01
Domain-Driven Design (DDD) software modeling delivers powerful results in practice, not just in theory, which is why developers worldwide are rapidly moving to adopt it. Now, for the first time, there’s an accessible guide to the basics of DDD: What it is, what problems it solves, how it works, and how to quickly gain value from it. Concise, readable, and actionable, *Domain-Driven Design Distilled* never buries you in detail—it focuses on what you need to know to get results. Vaughn Vernon, author of the best-selling *Implementing Domain-Driven Design*, draws on his twenty years of experience applying DDD principles to real-world situations. He is uniquely well-qualified to demystify its complexities, illuminate its subtleties, and help you solve the problems you might encounter. Vernon guides you through each core DDD technique for building better software. You’ll learn how to segregate domain models using the powerful Bounded Contexts pattern, to develop a Ubiquitous Language within an explicitly bounded context, and to help domain experts and developers work together to create that language. Vernon shows how to use Subdomains to handle legacy systems and to integrate multiple Bounded Contexts to define both team relationships and technical mechanisms. *Domain-Driven Design Distilled* brings DDD to life. Whether you’re a developer, architect, analyst, consultant, or customer, Vernon helps you truly understand it so you can benefit from its remarkable power. Coverage includes What DDD can do for you and your organization—and why it’s so important The cornerstones of strategic design with DDD: Bounded Contexts and Ubiquitous Language Strategic design with Subdomains Context Mapping: helping teams work together and integrate software more strategically Tactical design with Aggregates and Domain Events Using project acceleration and management tools to establish and maintain team cadence

Programming Pearls - Jon Bentley 2016-04-21
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.

Coders at Work - Peter Seibel 2009-12-21
Peter Seibel interviews 15 of the most interesting computer programmers alive today in *Coders at Work*, offering a companion volume to Apress’s highly acclaimed best-seller *Founders at Work* by Jessica Livingston. As the words “at work” suggest, Peter Seibel focuses on how his interviewees tackle the day-to-day work of programming, while revealing much more, like how they became great programmers, how they recognize programming talent in others, and what kinds of problems they find most interesting. Hundreds of people have suggested names of programmers to interview on the *Coders at Work* web site: www.codersatwork.com. The complete list was 284 names. Having digested everyone’s feedback, we selected 15 folks who’ve been kind enough to agree to be interviewed: Frances Allen: Pioneer in optimizing compilers, first woman to win the Turing Award (2006) and first female IBM fellow Joe Armstrong: Inventor of Erlang Joshua Bloch: Author of the Java collections framework, now at Google Bernie Cosell: One of the main software guys behind the original ARPANET IMPs and a master debugger Douglas Crockford: JSON founder, JavaScript architect at Yahoo! L. Peter Deutsch: Author of Ghostscript, implementer of Smalltalk-80 at Xerox PARC and Lisp 1.5 on PDP-1 Brendan Eich: Inventor of JavaScript, CTO of the Mozilla Corporation Brad Fitzpatrick: Writer of LiveJournal, OpenID, memcached, and Perlbal Dan Ingalls: Smalltalk implementor and designer Simon Peyton Jones: Coinventor of Haskell and lead designer of Glasgow Haskell Compiler Donald Knuth: Author of *The Art of Computer Programming* and creator of TeX Peter Norvig: Director of Research at Google and author of the standard text on AI Guy Steele: Coinventor of Scheme and part of the Common Lisp Gang of Five, currently working on Fortress Ken Thompson: Inventor of UNIX Jamie Zawinski: Author of XEmacs and early Netscape/Mozilla hacker

Developing Information Systems - Tahir Ahmed 2014
Systems development is the process of creating and maintaining information systems, including hardware, software, data, procedures and people. It combines technical expertise with business knowledge and management skill. This practical book provides a comprehensive introduction to the topic and can also be used as a handy reference guide. It discusses key elements of systems development and is the only textbook that supports the BCS Certificate in Systems Development.

Founders at Work - Jessica Livingston 2008-11-01
Now available in paperback—with a new preface and interview with Jessica Livingston about Y Combinator! *Founders at Work: Stories of Startups' Early Days* is a collection of interviews with founders of famous technology companies about what happened in the very earliest days. These people are celebrities now. What was it like when they were just a couple friends with an idea? Founders like Steve Wozniak (Apple), Caterina Fake (Flickr), Mitch Kapor (Lotus), Max Levchin (PayPal), and Sabeer Bhatia (Hotmail) tell you in their own words about their surprising and often very funny discoveries as they learned how to build a company. Where did they get the ideas that made them rich? How did they convince investors to back them? What went wrong, and how did they recover? Nearly all technical people have thought of one day starting or working for a startup. For them, this book is the closest you can come to being a fly on the wall at a successful startup, to learn how it's done. But ultimately these interviews are required reading for anyone who wants to understand business, because startups are business reduced to its essence. The reason their founders become rich is that startups do what businesses do—create value—more intensively than almost any other part of the economy. How? What are the secrets that make successful startups so insanely productive? Read this book, and let the founders themselves tell you.

Rules of Play - Katie Salen Tekinbas 2003-09-25
An impassioned look at games and game design that offers the most ambitious framework for understanding them to date. As pop culture, games are as important as film or television—but game design has yet to develop a theoretical framework or critical vocabulary. In *Rules of Play* Katie Salen and Eric Zimmerman present a much-needed primer for this emerging field. They offer a unified model for looking at all kinds of games, from board games and sports to computer and video games. As active participants in game culture, the authors have written *Rules of Play* as a catalyst for innovation, filled with new concepts, strategies, and

methodologies for creating and understanding games. Building an aesthetics of interactive systems, Salen and Zimmerman define core concepts like "play," "design," and "interactivity." They look at games through a series of eighteen "game design schemas," or conceptual frameworks, including games as systems of emergence and information, as contexts for social play, as a storytelling medium, and as sites of cultural resistance. Written for game scholars, game developers, and interactive designers, *Rules of Play* is a textbook, reference book, and theoretical guide. It is the first comprehensive attempt to establish a solid theoretical framework for the emerging discipline of game design.

More Programming Pearls - Jon Louis Bentley 1988

Software -- Software Engineering.

Code Complete, 2nd Edition - Steve McConnell

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.

Coders - Clive Thompson 2020-03-24

Facebook's algorithms shaping the news. Self-driving cars roaming the streets. Revolution on Twitter and romance on Tinder. We live in a world constructed of code--and coders are the ones who built it for us. Programmers shape our everyday behavior: When they make something easy to do, we do more of it. When they make it hard or impossible, we do less of it. From acclaimed tech writer Clive Thompson comes a

brilliant anthropological reckoning with the most powerful tribe in the world today, computer programmers, in a book that interrogates who they are, how they think, what qualifies as greatness in their world, and what should give us pause. In pop culture and media, the people who create the code that rules our world are regularly portrayed in hackneyed, simplified terms, as ciphers in hoodies. Thompson goes far deeper, taking us close to some of the great programmers of our time, including the creators of Facebook's News Feed, Instagram, Google's cutting-edge AI, and more. Speaking to everyone from revered "10X" elites to neophytes, back-end engineers and front-end designers, Thompson explores the distinctive psychology of this vocation--which combines a love of logic, an obsession with efficiency, the joy of puzzle-solving, and a superhuman tolerance for mind-bending frustration. Along the way, *Coders* ponders the morality and politics of code, including its implications for civic life and the economy and the major controversies of our era. In accessible, erudite prose, Thompson unpacks the surprising history of the field, beginning with the first coders -- brilliant and pioneering women, who, despite crafting some of the earliest personal computers and programming languages, were later written out of history. At the same time, the book deftly illustrates how programming has become a marvelous new art form--a source of delight and creativity, not merely danger. To get as close to his subject as possible, Thompson picks up the thread of his own long-abandoned coding skills as he reckons, in his signature, highly personal style, with what superb programming looks like. To understand the world today, we need to understand code and its consequences. With *Coders*, Thompson gives a definitive look into the heart of the machine.

Hackers & Painters - Paul Graham 2004-05-18

The author examines issues such as the rightness of web-based applications, the programming language renaissance, spam filtering, the Open Source Movement, Internet startups and more. He also tells important stories about the kinds of people behind technical innovations, revealing their character and their craft.