
The Self Taught Programmer The Definitive Guide To Programming Professionally

The Self Taught Programmer The Definitive Guide To Programming Professionally

Downloaded from staging.editor.seenic.io by Jonas & Aguirre

THE SELF TAUGHT PROGRAMMER THE DEFINITIVE GUIDE TO PROGRAMMING PROFESSIONALLY SUMMARY COLLECTION: UNLOCK THE SIGNIFICANCE IN BITE-SIZED CHUNKS

Welcome to our exciting book recap collection. We are delighted to introduce you to the globe of The Self Taught Programmer The Definitive Guide To Programming Professionally recaps and how they can improve your analysis experience. As serious visitors ourselves, we comprehend the worth of diving right into the heart of every story and discovering its significance in bite-sized chunks.

The Self Taught Programmer The Definitive Guide To Programming Professionally publication summary collection provides just that - a succinct and insightful summary of the key points and motifs of a publication. In today's hectic world, we know that time is precious, and our recaps are made to save you time by providing a fast introduction of The Self Taught Programmer The Definitive Guide To Programming Professionally's web content and insights.

Our team of expert authors carefully curates our publication recap of The Self Taught Programmer The Definitive Guide To Programming Professionally collection to ensure that we give you with high-quality recaps that record the essence of each publication. Whether you are aiming to check out new categories, find new writers, or merely obtain deeper understandings into your preferred publications, our collection has something for everybody.

Join us today and unlock the world of The Self Taught Programmer The Definitive Guide To Programming Professionally summaries. Discover the benefits of condensing complex concepts right into straightforward and easy-to-understand language. Our publication recaps are a great means to expand your expertise and expand your perspectives without having to spend hours of your time.

Remain tuned as we discover the principle of The Self Taught Programmer The Definitive Guide To Programming Professionally, discuss their benefits, and offer pointers on exactly how to create effective summaries. With our assistance, you'll find the right book for your interests and unlock a globe of understanding.

CHECKING OUT BOOK RECAPS OF THE SELF TAUGHT PROGRAMMER THE DEFINITIVE GUIDE TO PROGRAMMING PROFESSIONALLY

At our book summary collection, we securely count on the power of checking out The Self Taught Programmer The Definitive Guide To Programming Professionally. Not just can this open up brand-new expertise and understandings, yet it can likewise conserve viewers time and assist them determine which publications to invest their time in. Allow's dive into the concept of The Self Taught Programmer The Definitive Guide To Programming Professionally recaps and their advantages.

What are book summaries?

Book summaries are compressed versions of a publication's bottom lines and styles. They supply a fast introduction of The Self Taught Programmer The Definitive Guide To Programming Professionally's significance in bite-sized pieces. They can vary from a few paragraphs to a couple of web pages.

Why are they important?

The Self Taught Programmer The Definitive Guide To Programming Professionally recaps are beneficial since they permit viewers to acquire a much deeper understanding of a book's bottom lines and themes without needing to read the complete publication. They are specifically useful for active people that wish to remain informed yet may not have the time to check out an entire publication of The Self Taught Programmer The Definitive Guide To Programming Professionally.

Exactly how can they benefit The Self Taught Programmer The Definitive Guide To Programming Professionally readers?

Book recaps can benefit visitors by conserving time, giving a hassle-free introduction of The Self Taught Programmer The Definitive Guide To Programming Professionally's essence, and assisting viewers identify which publications deserve spending even more time in. They permit readers to rapidly and quickly gain understandings and expertise without needing to dedicate to reviewing the full book of The Self Taught Programmer The Definitive Guide To Programming Professionally.

- Conserves time
- Provides a fast overview
- Assists The Self Taught Programmer The Definitive Guide To Programming Professionally viewers decide which publications to spend more time in

Keep tuned for our next section where we will dive deeper into the benefits of The Self Taught Programmer The Definitive Guide To Programming Professionally.

ADVANTAGES OF THE SELF TAUGHT PROGRAMMER THE DEFINITIVE GUIDE TO PROGRAMMING PROFESSIONALLY PUBLICATION SUMMARIES

At our book summary collection, our team believe in the numerous advantages of checking out The Self Taught Programmer The Definitive Guide To Programming Professionally recaps. Below are a few crucial benefits:

- **Time-saving:** With our active schedules, it can be testing to find time to check out every book we desire. Our book summaries provide a quick overview of the most essential points without requiring to invest numerous hours in reviewing The Self Taught Programmer The Definitive Guide To Programming Professionally entire publication.
- **Quick overview of The Self Taught Programmer The Definitive Guide To Programming Professionally:** If there is a publication you want, however you're not sure if it's appropriate for you, our publication summaries provide a glimpse right into the author's main points and composing design prior to purchasing the full book.
- **Enhanced understanding in The Self Taught Programmer The Definitive Guide To Programming Professionally:** For those who have checked out the whole book, our book summaries use a chance to freshen your memory and discover the key points and themes.

Overall, book recaps of The Self Taught Programmer The Definitive Guide To Programming Professionally deal a beneficial device to boost your reading experience and maximize your time and effort.

EXACTLY HOW TO COMPOSE A BOOK SUMMARY OF THE SELF TAUGHT PROGRAMMER THE DEFINITIVE GUIDE TO PROGRAMMING PROFESSIONALLY

Creating a book summary may seem like a complicated task, but it can in fact be a fun and gratifying experience. Here are some crucial elements to remember when writing your publication summary:

1. **Focus on the essence:** The goal of a book summary is to catch the essence of The Self Taught Programmer The Definitive Guide To Programming Professionally in a concise and compelling way. Stay clear of getting captured up in the details and instead focus on the bottom lines and motifs that the writer is attempting to convey.
2. **Keep it short:** The Self Taught Programmer The Definitive Guide To Programming Professionally summary is implied to be a fast summary, so maintain it succinct. Adhere to the most essential information and stay clear of entering into excessive deepness.
3. **Consist of the major characters:** Make sure to include a brief summary of the major personalities, including their names and any type of defining characteristics or attributes.
4. **Highlight the main themes:** Identify the main styles of The Self Taught Programmer The Definitive Guide To Programming Professionally and highlight them in your summary. This will certainly give viewers a better idea of what the book has to do with and what they can expect to pick up from it.

By keeping these crucial elements in mind, you can compose an effective and engaging book summary that captures the significance of The Self Taught Programmer The Definitive Guide To Programming Professionally book and leaves visitors wanting a lot more.

FINDING THE RIGHT THE SELF TAUGHT PROGRAMMER THE DEFINITIVE GUIDE TO PROGRAMMING PROFESSIONALLY PUBLICATION SUMMARIES

Are you having a hard time to locate the right The Self Taught Programmer The Definitive Guide To Programming Professionally recaps for your passions? Do not stress, we've got you covered. Below are some suggestions on finding top quality book recaps:

1. Online Operating systems

One of the simplest means to discover The Self Taught Programmer The Definitive Guide To Programming Professionally recaps is through online systems. Websites like Blinkist, getAbstract, and Sumizeit offer a variety of summaries for various groups and styles. You can also look into Amazon Kindle's "Brief Reads" section for quick, easy-to-digest summaries.

2. Book Evaluation Internet Sites

Schedule review web sites like Goodreads and BookPage often feature summaries alongside their testimonials. They can give a much deeper understanding of The Self Taught Programmer The Definitive Guide To Programming Professionally story and themes while also supplying insight right into the reader's experience. You can also look into their "suggested" web page to find brand-new recaps.

3. Curated Collections

For viewers who choose a much more customized touch, curated collections are a wonderful option. These collections are frequently created by industry professionals or lovers and give a listing of must-read recaps for different styles. You can find them on blog sites, podcasts, and even social networks teams.

With these pointers, you can find the right The Self Taught Programmer The Definitive Guide To Programming Professionally publication recaps for your passions and choices. Delighted analysis!

The Self-taught Programmer

The Definitive Guide to Programming Professionally *Hachette UK*

'One of the best software design books of all time' - BookAuthority Cory Althoff is a self-taught programmer. After a year of self-study, he learned to program well enough to land a job as a software engineer II at eBay. But once he got there, he realised he was severely under-prepared. He was overwhelmed by the amount of things he needed to know but hadn't learned. His journey learning to program, and his experience in first software engineering job were the inspiration for this book. This book is not just about learning to program, although you will learn to code. If you want to program professionally, it is not enough to learn to code; that is why, in addition to helping you learn to program, Althoff also cover the rest of the things you need to know to program professionally that classes and books don't teach you. The Self-taught Programmer is a roadmap, a guide to take you from writing your first Python program to passing your first technical interview. The book is divided into five sections: 1. Learn to program in Python 3 and build your first program. 2. Learn object-oriented programming and create a powerful Python program to get you hooked. 3. Learn to use tools like Git, Bash and regular expressions. Then use your new coding skills to build a web scraper. 4. Study computer science fundamentals like data structures and algorithms. 5. Finish with best coding practices, tips for working with a team and advice on landing a programming job. You can learn to program professionally. The path is there. Will you take it? From the author I spent one year writing The Self-Taught Programmer. It was an exciting and rewarding experience. I treated my book like a software project. After I finished writing it, I created a program to pick out all of the code examples from the book and execute them in Python to make sure all 300+ examples worked properly. Then I wrote software to add line numbers and color to every code example. Finally, I had a group of 200 new programmers 'beta read' the book to identify poorly explained concepts and look for any errors my program missed. I hope you learn as much reading my book as I did writing it. Best of luck with your programming!

The Self-Taught Programmer

The Definitive Guide to Programming Professionally *Robinson*

The Self-Taught Computer Scientist

The Beginner's Guide to Data Structures & Algorithms *John Wiley & Sons*

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.

The Self-Taught Computer Scientist

The Beginner's Guide to Computer Science *John Wiley & Sons*

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.

Classic Computer Science Problems in Java *Simon and Schuster*

Sharpen your coding skills by exploring established computer science problems! Classic Computer Science Problems in Java challenges you with time-tested scenarios and algorithms. Summary Sharpen your coding skills by exploring established computer science problems! Classic Computer Science Problems in Java challenges you with time-tested scenarios and algorithms. You'll work through a series of exercises based in computer science fundamentals that are designed to improve your software development abilities, improve your understanding of artificial intelligence, and even prepare you to ace an interview. As you work through examples in search, clustering, graphs, and more, you'll remember important things you've forgotten and discover classic solutions to your "new" problems! Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the technology Whatever software development problem you're facing, odds are someone has already uncovered a solution. This book collects the most useful solutions devised, guiding you through a variety of challenges and tried-and-true problem-solving techniques. The principles and algorithms presented here are guaranteed to save you countless hours in project after project. About the book Classic Computer Science Problems in Java is a master class in computer programming designed around 55 exercises that have been used in computer science classrooms for years. You'll work through hands-on examples as you explore core algorithms, constraint problems, AI applications, and much more. What's inside Recursion, memoization, and bit manipulation Search, graph, and genetic algorithms Constraint-satisfaction problems K-means clustering, neural networks, and adversarial search About the reader For intermediate Java programmers. About the author David Kopec is an assistant professor of Computer Science and Innovation at Champlain College in Burlington, Vermont. Table of Contents 1 Small problems 2 Search problems 3 Constraint-satisfaction problems 4 Graph problems 5 Genetic algorithms 6 K-means clustering 7 Fairly simple neural networks 8 Adversarial search 9 Miscellaneous problems 10 Interview with Brian Goetz

A Programmer's Guide to Computer Science

A Virtual Degree for the Self-taught Developer

You know how to code..but is it enough? Do you feel left out when other programmers talk about asymptotic bounds? Have you failed a job interview because you don't know computer science? The author, a senior developer at a major software company with a PhD in computer science, takes you through what you would have learned while earning a four-year computer science degree. Volume one covers the most frequently referenced topics, including algorithms and data structures, graphs, problem-solving techniques, and complexity theory. When you finish this book, you'll have the tools you need to hold your own with people who have - or expect you to have - a computer science degree.

Python Crash Course

A Hands-On, Project-Based Introduction to Programming *No Starch Press*

Python Crash Course is a fast-paced, thorough introduction to Python that will have you writing programs, solving problems, and making things that work in no time. In the first half of the book, you'll learn about basic programming concepts, such as lists, dictionaries, classes, and loops, and practice writing clean and readable code with exercises for each topic. You'll also learn how to make your programs interactive and how to test your code safely before adding it to a project. In the second half of the book, you'll put your new knowledge into practice with three substantial projects: a Space Invaders-inspired arcade game, data visualizations with Python's super-handY libraries, and a simple web app you can deploy online. As you work through Python Crash Course you'll learn how to: -Use powerful Python libraries and tools, including matplotlib, NumPy, and Pygal -Make 2D games that respond to keypresses and mouse clicks, and that grow more difficult as the game progresses -Work with data to generate interactive visualizations -Create and customize Web apps and deploy them safely online -Deal with mistakes and errors so you can solve your own programming problems If you've been thinking seriously about digging into programming, Python Crash Course will get you up to speed and have you writing real programs fast. Why wait any longer? Start your engines and code! Uses Python 2 and 3

The Self-Taught Developer *Tommy Chheng*

Want a career as a software engineer? Don't want to spend years or the money going to school? Have to write code for your current job? The lessons in this book are all things author Tommy Chheng learned are vital to developers during his career. This book will teach you: * What tools you will need * How to ask the right questions * How to solve a programming problem * The important Computer Science topics * How to get hired

The Self-Taught Programmer

The Definitive Guide to Programming Professionally

I am a self-taught programmer. After a year of self-study, I learned to program well enough to land a job as a software engineer II at eBay. Once I got there, I realized I was severely under-prepared. I was overwhelmed by the amount of things I needed to know but hadn't learned yet. My journey learning to program, and my experience at my first job as a software engineer were the inspiration for this book. This book is not just about learning to program; although you will learn to code. If you want to program professionally, it is not enough to learn to code; that is why, in addition to helping

you learn to program, I also cover the rest of the things you need to know to program professionally that classes and books don't teach you. "The Self-taught Programmer" is a roadmap, a guide to take you from writing your first Python program, to passing your first technical interview. I divided the book into six sections: 1. Learn to program in Python 3 and build your first program. 2. Learn Object-oriented programming and create a powerful Python program to get you hooked. 3. Learn to use tools like Git, Bash, regular expressions and databases. Then use your new coding skills to build a web scraper. 4. Study Computer Science fundamentals including computer architecture, data structures, algorithms and network programming. 5. Learn to program for production: I cover the software development process, testing, and best coding practices.6. Finish with tips for working with a team and landing a programming job.You CAN learn to program professionally. The path is there. Will you take it?

Software Engineering as a Career

How to Land a Programming Job Without a Computer Science Degree, Habits of Successful Self-Taught Coders and Avoiding Programmer Burnout *Independently Published*

Starting a career as a software engineer without a computer science degree is a long and difficult journey, Hasan Armstrong discovered this whilst attempting to switch from a career in healthcare to software engineering. He now works as a software engineer and incorporates all the lessons he has learnt in this book. This book will provide a roadmap to getting a job as a software engineer without a computer science degree, as well as providing solutions to the obstacles you may face along the way, like learning new programming languages, handling interview questions, negotiating job offers and much more. Through his youtube channel, Hasan has helped several thousands of people learn to code. What you will learn in this book? How to determine if a job as a software engineer is even for you? Should you become a front-end, backend or full stack software engineer? Mindsets and habits of software engineers who seek excellence. Programming topics you will need to learn and practice before you can start applying for software engineering roles. Practices to stay healthy, avoid burnout syndrome and remain happy and fulfilled as a self-taught software engineer. Increase the likelihood of landing a software engineering role, by creating a personal brand, a CV that stands out and finding companies you want to work for. Mindsets and habits of exceptional software engineers Interviewer asks "What kind of salary do you expect for this role?" - How should you reply? You've started working as a software engineer. How can you climb the career ladder? The dark side of working as a software engineer. How should you handle workplace politics, mental health issues and technical debt? We are keen to help you land a software engineering role and help you progress in that role. So if you want to know if software engineering is for you, in the process of learning to code or applying for software engineering roles this book is worth purchasing. **Buy the paperback version of this book, and get the kindle version absolutely FREE**

Team Geek

A Software Developer's Guide to Working Well with Others *"O'Reilly Media, Inc."*

In a perfect world, software engineers who produce the best code are the most successful. But in our perfectly messy world, success also depends on how you work with people to get your job done. In this highly entertaining book, Brian Fitzpatrick and Ben Collins-Sussman cover basic patterns and anti-patterns for working with other people, teams, and users while trying to develop software. This is valuable information from two respected software engineers whose popular series of talks—including "Working with Poisonous People"—has attracted hundreds of thousands of followers. Writing software is a team sport, and human factors have as much influence on the outcome as technical factors. Even if you’ve spent decades learning the technical side of programming, this book teaches you about the often-overlooked human component. By learning to collaborate and investing in the "soft skills" of software engineering, you can have a much greater impact for the same amount of effort. Team Geek was named as a Finalist in the 2013 Jolt Awards from Dr. Dobb's Journal. The publication's panel of judges chose five notable books, published during a 12-month period ending June 30, that every serious programmer should read.

Coding All-in-One For Dummies *John Wiley & Sons*

See all the things coding can accomplish The demand for people with coding know-how exceeds the number of people who understand the languages that power technology. Coding All-in-One For Dummies gives you an ideal place to start when you're ready to add this valuable asset to your professional repertoire. Whether you need to learn how coding works to build a web page or an application or see how coding drives the data revolution, this resource introduces the languages and processes you'll need to know. Peek inside to quickly learn the basics of simple web languages, then move on to start thinking like a professional coder and using languages that power big applications. Take a look inside for the steps to get started with updating a website, creating the next great mobile app, or exploring the world of data science. Whether you're looking for a complete beginner's guide or a trusted resource for when you encounter problems with coding, there's something for you! Create code for the web Get the tools to create a mobile app Discover languages that power data science See the future of coding with machine learning tools With the demand for skilled coders at an all-time high, Coding All-in-One For Dummies is here to propel coding newbies to the ranks of professional programmers.

A Programmer's Guide to Computer Science Vol. 2 *Jaxson Media*

You know how to code... ...but is it enough?Do you feel left out when other programmers talk about asymptotic bounds?Have you failed a job interview because you don't know computer science?Volume two picks up where volume one left off, covering proofs, security, hardware and software, and various advanced topics.You've learned the basics. Are you ready for what comes next?

Beginner's Step-by-Step Coding Course *Penguin*

With this visual guide to computer programming for beginners, it has never been easier to learn how to code. Coding skills are in high demand and

the need for programmers is still growing. Covering three of the most popular languages for new coders, this book uses a graphic method to break complex subjects into user-friendly chunks, bringing essential skills within easy reach. Each chapter contains tutorials on practical projects designed to teach you the main applications of each language, such as building websites, creating games, and designing apps. The book also looks at many of the main coding languages that are out there, outlining the key applications of each language, so you can choose the right language for you. You'll learn to think like a programmer by breaking a problem down into parts, before turning those parts into lines of code. Short, easy-to-follow steps then show you, piece by piece, how to build a complete program. There are challenges for you to tackle to build your confidence before moving on. Written by a team of expert coders and coding teachers, Beginner's Step-by-Step Coding Course is the ideal way to get to set you on the road to code.

Late Bloomers

The Hidden Strengths of Learning and Succeeding at Your Own Pace *Broadway Books*

A groundbreaking exploration of what it means to be a late bloomer in a culture obsessed with SAT scores and early success, and how finding one's way later in life can be an advantage to long-term achievement and happiness. We live in a society where kids and parents are obsessed with early achievement, from getting perfect scores on SATs to getting into Ivy League colleges to landing an amazing job at Google or Facebook--or even better, creating a startup with the potential to be the next Google or Facebook or Uber. We see software coders becoming millionaires or even billionaires before age 30 and feel we are failing if we are not one of them. But there is good news. A lot of us--most of us--do not explode out of the gates in life. That was true for author Rich Karlgaard, who had a mediocre academic career at Stanford (which he got into by a fluke), and after graduating, worked as a dishwasher, nightwatchman, and typing temp before finally finding the inner motivation and drive that ultimately led him to start up a high-tech magazine in Silicon Valley, and eventually to become the publisher of Forbes magazine. There is a scientific explanation for why so many of us bloom later in life. The executive function of our brains doesn't mature until age 25--and later for some. In fact our brain's capabilities peak at different ages. We actually enjoy multiple periods of blooming in our lives. Based on several years of research, personal experience, and interviews with neuroscientists and psychologists, and countless people at different stages of their careers, Bloom reveals how and when we achieve full potential--and why an algorithmic acuity in math is such an anomaly in terms of career success.

The Definitive Guide to Linux Network Programming *Apress*

* Clear and abundant examples, using real-world code, written by three experienced developers who write networking code for a living. * Describes how to build clients and servers, explains how TCP, UDP, and IP work, and shows how to debug networking applications via packet sniffing and deconstruction. * Well suited for Windows developer looking to expand to Linux, or for the proficient Linux developer looking to incorporate client-server programming into their application.

The Complete Software Developer's Career Guide *Simple Programmer, LLC*

"Early in his software developer career, John Sonmez discovered that technical knowledge alone isn't enough to break through to the next income level - developers need "soft skills" like the ability to learn new technologies just in time, communicate clearly with management and consulting clients, negotiate a fair hourly rate, and unite teammates and coworkers in working toward a common goal. Today John helps more than 1.4 million programmers every year to increase their income by developing this unique blend of skills. Who Should Read This Book? Entry-Level Developers - This book will show you how to ensure you have the technical skills your future boss is looking for, create a resume that leaps off a hiring manager's desk, and escape the "no work experience" trap. Mid-Career Developers - You'll see how to find and fill in gaps in your technical knowledge, position yourself as the one team member your boss can't live without, and turn those dreaded annual reviews into chance to make an iron-clad case for your salary bump. Senior Developers - This book will show you how to become a specialist who can command above-market wages, how building a name for yourself can make opportunities come to you, and how to decide whether consulting or entrepreneurship are paths you should pursue. Brand New Developers - In this book you'll discover what it's like to be a professional software developer, how to go from "I know some code" to possessing the skills to work on a development team, how to speed along your learning by avoiding common beginner traps, and how to decide whether you should invest in a programming degree or 'bootcamp.'"

Coders at Work

Reflections on the Craft of Programming *Apress*

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 Perlbai 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

Becoming a Better Programmer

A Handbook for People Who Care About Code *"O'Reilly Media, Inc."*

If you're passionate about programming and want to get better at it, you've come to the right source. Code Craft author Pete Goodliffe presents a collection of useful techniques and approaches to the art and craft of programming that will help boost your career and your well-being. Goodliffe presents sound advice that he's learned in 15 years of professional programming. The book's standalone chapters span the range of a software developer's life—dealing with code, learning the trade, and improving performance—with no language or industry bias. Whether you're a seasoned developer, a neophyte professional, or a hobbyist, you'll find valuable tips in five independent categories: Code-level techniques for crafting lines of code, testing, debugging, and coping with complexity Practices, approaches, and attitudes: keep it simple, collaborate well, reuse, and create malleable code Tactics for learning effectively, behaving ethically, finding challenges, and avoiding stagnation Practical ways to complete things: use the right tools, know what “done” looks like, and seek help from colleagues Habits for working well with others, and pursuing development as a social activity

The Imposter's Handbook

A CS Primer for Self-Taught Developers *Independently Published*

Don't have a CS degree? Neither does Rob. That's why he wrote this book: to fill the gaps in his career. The result? Over 450 pages of essentials skills and ideas every developer should know with illustrations by the author, who loves to sketch. An illustrated CS Primer, if you will. Rob is a self-taught software developer (like so many) and for most of his career he learned what was required to get the job done. When conversations veered toward core concepts, he disengaged. Rob decided to change all of this in 2014. He sat down and looked up all of the topics that a typical CS degree covers and then dove in. Half way through, he decided to write a book about what he was learning. That book is The Imposter's Handbook, a compendium of useful programming concepts from Algorithms to Complexity Theory, TDD to Garbage Collection. Things you should really know if you're paid to write software.

Sams Teach Yourself Beginning Programming in 24 Hours *Sams Publishing*

Sams Teach Yourself Beginning Programming in 24 Hours, Second Edition explains the basics of programming in the successful 24-Hours format. The book begins with the absolute basics of programming: Why program? What tools to use? How does a program tell the computer what to do? It teaches readers how to program the computer and then moves on by exploring the some most popular programming languages in use. The author starts by introducing the reader to the Basic language and finishes with basic programming techniques for Java, C++, and others.

Head First Python

A Brain-Friendly Guide *"O'Reilly Media, Inc."*

Want to learn the Python language without slogging your way through how-to manuals? With Head First Python, you'll quickly grasp Python's fundamentals, working with the built-in data structures and functions. Then you'll move on to building your very own webapp, exploring database management, exception handling, and data wrangling. If you're intrigued by what you can do with context managers, decorators, comprehensions, and generators, it's all here. This second edition is a complete learning experience that will help you become a bonafide Python programmer in no time. Why does this book look so different? Based on the latest research in cognitive science and learning theory, Head First Pythonuses a visually rich format to engage your mind, rather than a text-heavy approach that puts you to sleep. Why waste your time struggling with new concepts? This multi-sensory learning experience is designed for the way your brain really works.

Why Good People Can't Get Jobs

The Skills Gap and What Companies Can Do About It *University of Pennsylvania Press*

Peter Cappelli confronts the myth of the skills gap and provides an actionable path forward to put people back to work. Even in a time of perilously high unemployment, companies contend that they cannot find the employees they need. Pointing to a skills gap, employers argue applicants are simply not qualified; schools aren't preparing students for jobs; the government isn't letting in enough high-skill immigrants; and even when the match is right, prospective employees won't accept jobs at the wages offered. In this powerful and fast-reading book, Peter Cappelli, Wharton management professor and director of Wharton's Center for Human Resources, debunks the arguments and exposes the real reasons good people can't get hired. Drawing on jobs data, anecdotes from all sides of the employer-employee divide, and interviews with jobs professionals, he explores the paradoxical forces bearing down on the American workplace and lays out solutions that can help us break through what has become a crippling employer-employee stand-off. Among the questions he confronts: Is there really a skills gap? To what extent is the hiring process being held hostage by automated software that can crunch thousands of applications an hour? What kind of training could best bridge the gap between employer expectations and applicant realities, and who should foot the bill for it? Are schools really at fault? Named one of HR Magazine's Top 20 Most

Influential Thinkers of 2011, Cappelli not only changes the way we think about hiring but points the way forward to rev America's job engine again.

Don't Teach Coding

Until You Read This Book *John Wiley & Sons*

The definitive resource for understanding what coding is, designed for educators and parents Even though the vast majority of teachers, parents, and students understand the importance of computer science in the 21st century, many struggle to find appropriate educational resources. Don't Teach Coding: Until You Read This Book fills a gap in current knowledge by explaining exactly what coding is and addressing why and how to teach the subject. Providing a historically grounded, philosophically sensitive description of computer coding, this book helps readers understand the best practices for teaching computer science to their students and their children. The authors, experts in teaching computer sciences to students of all ages, offer practical insights on whether coding is a field for everyone, as opposed to a field reserved for specialists. This innovative book provides an overview of recent scientific research on how the brain learns coding, and features practical exercises that strengthen coding skills. Clear, straightforward chapters discuss a broad range of questions using principles of computer science, such as why we should teach students to code and is coding a science, engineering, technology, mathematics, or language? Helping readers understand the principles and issues of coding education, this book: Helps those with no previous background in computer science education understand the questions and debates within the field Explores the history of computer science education and its influence on the present Views teaching practices through a computational lens Addresses why many schools fail to teach computer science adequately Explains contemporary issues in computer science such as the language wars and trends that equate coding with essential life skills like reading and writing Don't Teach Coding: Until You Read This Book is a valuable resource for K-12 educators in computer science education and parents wishing to understand the field to help chart their children's education path.

Ultralearning

Master Hard Skills, Outsmart the Competition, and Accelerate Your Career *HarperCollins*

Learn a new talent, stay relevant, reinvent yourself, and adapt to whatever the workplace throws your way. Ultralearning offers nine principles to master hard skills quickly. This is the essential guide to future-proof your career and maximize your competitive advantage through self-education. In these tumultuous times of economic and technological change, staying ahead depends on continual self-education—a lifelong mastery of fresh ideas, subjects, and skills. If you want to accomplish more and stand apart from everyone else, you need to become an ultralearner. The challenge of learning new skills is that you think you already know how best to learn, as you did as a student, so you rerun old routines and old ways of solving problems. To counter that, Ultralearning offers powerful strategies to break you out of those mental ruts and introduces new training methods to help you push through to higher levels of retention. Scott H. Young incorporates the latest research about the most effective learning methods and the stories of other ultralearners like himself—among them Benjamin Franklin, chess grandmaster Judit Polgár, and Nobel laureate physicist Richard Feynman, as well as a host of others, such as little-known modern polymath Nigel Richards, who won the French World Scrabble Championship—without knowing French. Young documents the methods he and others have used to acquire knowledge and shows that, far from being an obscure skill limited to aggressive autodidacts, ultralearning is a powerful tool anyone can use to improve their career, studies, and life. Ultralearning explores this fascinating subculture, shares a proven framework for a successful ultralearning project, and offers insights into how you can organize and exe - cute a plan to learn anything deeply and quickly, without teachers or budget-busting tuition costs. Whether the goal is to be fluent in a language (or ten languages), earn the equivalent of a college degree in a fraction of the time, or master multiple tools to build a product or business from the ground up, the principles in Ultralearning will guide you to success.

Make Your Own Neural Network *Createspace Independent Publishing Platform*

A step-by-step gentle journey through the mathematics of neural networks, and making your own using the Python computer language. Neural networks are a key element of deep learning and artificial intelligence, which today is capable of some truly impressive feats. Yet too few really understand how neural networks actually work. This guide will take you on a fun and unhurried journey, starting from very simple ideas, and gradually building up an understanding of how neural networks work. You won't need any mathematics beyond secondary school, and an accessible introduction to calculus is also included. The ambition of this guide is to make neural networks as accessible as possible to as many readers as possible - there are enough texts for advanced readers already! You'll learn to code in Python and make your own neural network, teaching it to recognise human handwritten numbers, and performing as well as professionally developed networks. Part 1 is about ideas. We introduce the mathematical ideas underlying the neural networks, gently with lots of illustrations and examples. Part 2 is practical. We introduce the popular and easy to learn Python programming language, and gradually builds up a neural network which can learn to recognise human handwritten numbers, easily getting it to perform as well as networks made by professionals. Part 3 extends these ideas further. We push the performance of our neural network to an industry leading 98% using only simple ideas and code, test the network on your own handwriting, take a privileged peek inside the mysterious mind of a neural network, and even get it all working on a Raspberry Pi. All the code in this has been tested to work on a Raspberry Pi Zero.

Python Crash Course, 2nd Edition

A Hands-On, Project-Based Introduction to Programming *No Starch Press*

The second edition of the best-selling Python book in the world (over 1 million copies sold!). A fast-paced, no-nonsense guide to programming in

Python. Updated and thoroughly revised to reflect the latest in Python code and practices. Python Crash Course is the world's best-selling guide to the Python programming language. This fast-paced, thorough introduction to programming with Python will have you writing programs, solving problems, and making things that work in no time. In the first half of the book, you'll learn basic programming concepts, such as variables, lists, classes, and loops, and practice writing clean code with exercises for each topic. You'll also learn how to make your programs interactive and test your code safely before adding it to a project. In the second half, you'll put your new knowledge into practice with three substantial projects: a Space Invaders-inspired arcade game, a set of data visualizations with Python's handy libraries, and a simple web app you can deploy online. As you work through the book, you'll learn how to:

- Use powerful Python libraries and tools, including Pygame, Matplotlib, Plotly, and Django
- Make 2D games that respond to keypresses and mouse clicks, and that increase in difficulty
- Use data to generate interactive visualizations
- Create and customize web apps and deploy them safely online
- Deal with mistakes and errors so you can solve your own programming problems

If you've been thinking about digging into programming, Python Crash Course will get you writing real programs fast. Why wait any longer? Start your engines and code!

Computer Programming Crash Course

7 Books in 1- Coding Languages for Beginners: C++, C#, SQL, Python, Data Science for Python, Raspberry Pi and Arduino. Teach Yourself to Code. Learn Faster *Computer DM-Academy*

-- 55% OFF For Bookstores! -- Are you looking for the PERFECT introduction into the world of coding? Want to uncover the secrets of Python, SQL, C++ and so much more? Are you looking for the ultimate guide to getting started with programming? Then this bundle is for you. Written with the beginner in mind, this incredible 7-in-1 book bundle brings you everything you need to know about programming. Packed with a ton of advice and step-by-step instructions on all the most popular and useful languages, you'll explore how even a complete beginner can get started with ease! Covering data science, Arduino, and even Raspberry pi, you'll learn the fundamentals of object-oriented programming, operators, variables, loops, classes, arrays, strings and so much more! Here's just a little of what you'll discover inside: Uncovering The Secrets of C++, C#, Python, SQL and More Breaking Down The Fundamentals of Data Science Understanding The Different Classes, Operations, and Data Types Fundamental Programming Skills That YOU Need To Know Tips and Tricks For Getting The Most out of Each Language The Best Strategies For Using Arduino and Raspberry Pi Common Errors and How To Troubleshoot Them And Much More! No matter your level of programming experience, this bundle uses step-by-step instructions and easy-to-follow advice so you can get the most out of programming. Explore these amazing languages, master the fundamentals of programming, and unleash your programming potential today! Buy it now and let your customers start their journey in programming!

Human Development Report 2015

Work for Human Development *United Nations*

This report takes a broad view of the link between work and human development. Work is a critical tool for economic growth and security, poverty reduction and gender equality. It enables full participation in society while affording people a sense of dignity and worth. Humans working together not only increase their material well-being, they also accumulate a wide body of knowledge that serves as the basis for cultures and civilizations. The report finds that work enhances human development when policies are taken to expand productive, remunerative and satisfying work opportunities. Workers' skills and potentials are enhanced, their well-being in terms of rights, safety and benefits are ensured with targeted interventions, and an agenda incorporating decent work, a new Social Contract and a Global Deal is pursued.

Grokking Algorithms

An illustrated guide for programmers and other curious people *Simon and Schuster*

Summary Grokking Algorithms is a fully illustrated, friendly guide that teaches you how to apply common algorithms to the practical problems you face every day as a programmer. You'll start with sorting and searching and, as you build up your skills in thinking algorithmically, you'll tackle more complex concerns such as data compression and artificial intelligence. Each carefully presented example includes helpful diagrams and fully annotated code samples in Python. Learning about algorithms doesn't have to be boring! Get a sneak peek at the fun, illustrated, and friendly examples you'll find in Grokking Algorithms on Manning Publications' YouTube channel. Continue your journey into the world of algorithms with Algorithms in Motion, a practical, hands-on video course available exclusively at Manning.com (www.manning.com/livevideo/algorithms-in-motion). Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology An algorithm is nothing more than a step-by-step procedure for solving a problem. The algorithms you'll use most often as a programmer have already been discovered, tested, and proven. If you want to understand them but refuse to slog through dense multipage proofs, this is the book for you. This fully illustrated and engaging guide makes it easy to learn how to use the most important algorithms effectively in your own programs. About the Book Grokking Algorithms is a friendly take on this core computer science topic. In it, you'll learn how to apply common algorithms to the practical programming problems you face every day. You'll start with tasks like sorting and searching. As you build up your skills, you'll tackle more complex problems like data compression and artificial intelligence. Each carefully presented example includes helpful diagrams and fully annotated code samples in Python. By the end of this book, you will have mastered widely applicable algorithms as well as how and when to use them. What's Inside Covers search, sort, and graph algorithms Over 400 pictures with detailed walkthroughs Performance trade-offs between algorithms Python-based code samples About the Reader This easy-to-read, picture-heavy introduction is suitable for self-taught programmers, engineers, or anyone who wants to brush up on algorithms. About the Author Aditya Bhargava is a Software Engineer with a dual background in Computer Science and Fine Arts. He blogs on programming at adit.io. Table of Contents Introduction to algorithms Selection sort Recursion Quicksort Hash tables Breadth-first

search Dijkstra's algorithm Greedy algorithms Dynamic programming K-nearest neighbors

Computer Programming Bible

A Step by Step Guide On How To Master From The Basics to Advanced of Python, C, C++, C#, HTML Coding Raspberry Pi3

Whether your incentive to learn about computer programming stems from interest, or it's because you want a better paying job, starting with the basics and working your way up is the most promising approach to take.

The Go Programming Language *Addison-Wesley Professional*

The Go Programming Language is the authoritative resource for any programmer who wants to learn Go. It shows how to write clear and idiomatic Go to solve real-world problems. The book does not assume prior knowledge of Go nor experience with any specific language, so you'll find it accessible whether you're most comfortable with JavaScript, Ruby, Python, Java, or C++. The first chapter is a tutorial on the basic concepts of Go, introduced through programs for file I/O and text processing, simple graphics, and web clients and servers. Early chapters cover the structural elements of Go programs: syntax, control flow, data types, and the organization of a program into packages, files, and functions. The examples illustrate many packages from the standard library and show how to create new ones of your own. Later chapters explain the package mechanism in more detail, and how to build, test, and maintain projects using the go tool. The chapters on methods and interfaces introduce Go's unconventional approach to object-oriented programming, in which methods can be declared on any type and interfaces are implicitly satisfied. They explain the key principles of encapsulation, composition, and substitutability using realistic examples. Two chapters on concurrency present in-depth approaches to this increasingly important topic. The first, which covers the basic mechanisms of goroutines and channels, illustrates the style known as communicating sequential processes for which Go is renowned. The second covers more traditional aspects of concurrency with shared variables. These chapters provide a solid foundation for programmers encountering concurrency for the first time. The final two chapters explore lower-level features of Go. One covers the art of metaprogramming using reflection. The other shows how to use the unsafe package to step outside the type system for special situations, and how to use the cgo tool to create Go bindings for C libraries. The book features hundreds of interesting and practical examples of well-written Go code that cover the whole language, its most important packages, and a wide range of applications. Each chapter has exercises to test your understanding and explore extensions and alternatives. Source code is freely available for download from <http://gopl.io/> and may be conveniently fetched, built, and installed using the go get command.

Python Flash Cards

Syntax, Concepts, and Examples

Self-Care for the Real World

Practical self-care advice for everyday life *Random House*

THE TOP TEN BESTSELLER 'Unusually practical, non-patronising and authentic. Think Marie Kondo for the mind' Sunday Times Wellness pioneers Nadia Narain and Katia Narain Phillips have spent decades helping others to feel their best. But it took them a bit longer to learn to care for themselves. Here they share the small, achievable steps they picked up on a lifetime's journey towards self-care, and how you can apply them to your life, wherever you are. Right now, you may be deep in the waves of life, being tossed around. Learning self-care is like building your own life boat, plank by plank. Once you've got your boat, you'll still be rocked by the same waves, but you'll have a feeling of safety, and a stability that means you can pick other people up on your way.

Scratch Coding Cards

Creative Coding Activities for Kids

A collection of ten themed activity card sets that introduces children to computer programming fundamentals using Scratch, a visual programming language developed by the Lifelong Kindergarten Group at the MIT Media Lab.

Computer Programming

The Most Complete Crash Course for Learning The Perfect Skills To Coding Your Project Even If You Are an Absolute Beginner. Learn and Master The Best Programming Languages *Francesco Cammardella*

When you start programming from scratch, you are faced with a difficult choice: which language to choose? What is the best language to start programming? Imagine learning to program in a language only to discover that it is not used by companies, or that it is in great demand, but not what you were planning to do. Maybe you dreamed of developing smartphone applications and instead you find yourself building websites! Wouldn't you like to be able to choose your first programming language having all this information at your disposal? This complete and exhaustive Manual will guide you in the computer programming world, introducing you to the best programs that will be useful also for developing your career, even if you are an absolute beginner. In this book you will:

- Learn Python and understand why it has consistently ranked in the top ten most popular programming languages. You will be able to master it at its best, as required by the business market today, no career will be precluded if you know

this extraordinary computer programming. - Understand How to Manipulate Data with SQL to query and modify database data. You will learn how to insert, update, and delete records from tables using SQL statements - Be Able To Create Games or Apps with CC+ or C#, to get the perfect coding skills to develop the video game of your dreams, both for your own entertainment or making some money out of it - Discover How To Use Raspberry Pi, the tiny computer that allows you to harness tech skills and let imaginations run wild. It enables people of all ages to explore computing and to learn how to program in languages like Scratch and Python. - Have the Chance to Practice What You Learn with many examples and exercises of projects and codes with step-by-step explanations accompanied by illustrative images and tables. - ... & Lot More! Deciding which computer programming language to learn isn't quite so straightforward; it all depends on what your goal is, what task you want to achieve, or what problem you need to solve. It may be more beneficial for your career to learn one of the "dreaded" computer programming languages, as you'll be more in demand. You can find and learn all these dreaded" computer programming languages in this definitive Crash Course that will make all of them easy to use and understand. You will get all the necessary skills to face the computer programming world at your best, even if you start from scratch, and will be so easy to do with this book in your hands that you will be asked yourself why you haven't start early! Don't keep on wasting your time... Order Your Copy Now and Start Coding Like a Pro!

Inside the Machine

An Illustrated Introduction to Microprocessors and Computer Architecture *No Starch Press*
Om hvordan mikroprocessorer fungerer, med undersøgelse af de nyeste mikroprocessorer fra Intel, IBM og Motorola.

Think Like a Programmer

An Introduction to Creative Problem Solving *No Starch Press*
The real challenge of programming isn't learning a language's syntax—it's learning to creatively solve problems so you can build something great. In this one-of-a-kind text, author V. Anton Spraul breaks down the ways that programmers solve problems and teaches you what other introductory books often ignore: how to Think Like a Programmer. Each chapter tackles a single programming concept, like classes, pointers, and recursion, and open-ended exercises throughout challenge you to apply your knowledge. You'll also learn how to: -Split problems into discrete components to make them easier to solve -Make the most of code reuse with functions, classes, and libraries -Pick the perfect data structure for a particular job -Master more advanced programming tools like recursion and dynamic memory -Organize your thoughts and develop strategies to tackle particular types of problems Although the book's examples are written in C++, the creative problem-solving concepts they illustrate go beyond any particular language; in fact, they often reach outside the realm of computer science. As the most skillful programmers know, writing great code is a creative art—and the first step in creating your masterpiece is learning to Think Like a Programmer.

Python for Kids

A Playful Introduction To Programming *No Starch Press*
Python is a powerful, expressive programming language that’s easy to learn and fun to use! But books about learning to program in Python can be

kind of dull, gray, and boring, and that’s no fun for anyone. Python for Kids brings Python to life and brings you (and your parents) into the world of programming. The ever-patient Jason R. Briggs will guide you through the basics as you experiment with unique (and often hilarious) example programs that feature ravenous monsters, secret agents, thieving ravens, and more. New terms are defined; code is colored, dissected, and explained; and quirky, full-color illustrations keep things on the lighter side. Chapters end with programming puzzles designed to stretch your brain and strengthen your understanding. By the end of the book you'll have programmed two complete games: a clone of the famous Pong and "Mr. Stick Man Races for the Exit"—a platform game with jumps, animation, and much more. As you strike out on your programming adventure, you'll learn how to: -Use fundamental data structures like lists, tuples, and maps -Organize and reuse your code with functions and modules -Use control structures like loops and conditional statements -Draw shapes and patterns with Python’s turtle module -Create games, animations, and other graphical wonders with tkinter Why should serious adults have all the fun? Python for Kids is your ticket into the amazing world of computer programming. For kids ages 10+ (and their parents) The code in this book runs on almost anything: Windows, Mac, Linux, even an OLPC laptop or Raspberry Pi!

The Pragmatic Programmer

your journey to mastery, 20th Anniversary Edition *Addison-Wesley Professional*
“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.