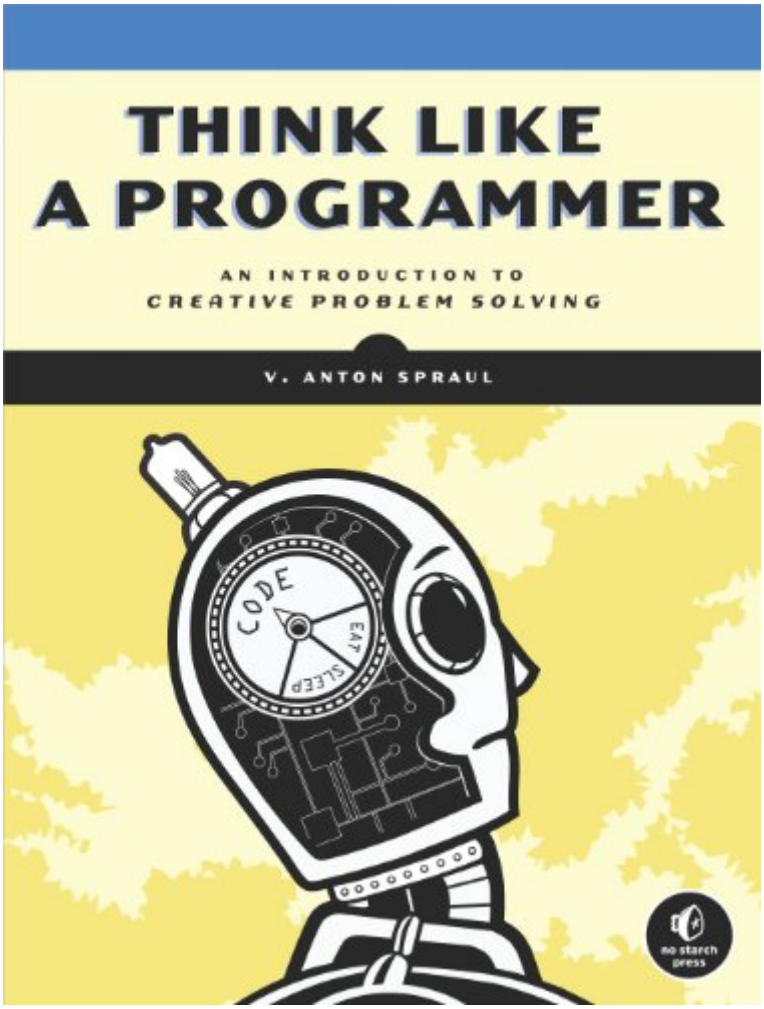
The book was found

Think Like A Programmer: An Introduction To Creative Problem Solving



PDF
Adobe
DOWNLOAD EBOOK

Synopsis

Rewire your brainExamples in this book are written in C++, but will improve your ability to think like a programmer in any language.The real challenge of programming isn't learning a language's syntaxâ "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 solveMake the most of code reuse with functions, classes, and librariesPick the perfect data structure for a particular jobMaster more advanced programming tools like recursion and dynamic memoryOrganize your thoughts and develop strategies to tackle particular types of problemsAlthough 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 artâ "and the first step in creating your masterpiece is learning to Think Like a Programmer.

Book Information

File Size: 2290 KB

Print Length: 260 pages

Page Numbers Source ISBN: 1593274246

Simultaneous Device Usage: Unlimited

Publisher: No Starch Press; 1 edition (August 1, 2012)

Publication Date: August 1, 2012

Sold by:Â Digital Services LLC

Language: English

ASIN: B008RQMDU4

Text-to-Speech: Enabled

X-Ray: Not Enabled

Word Wise: Not Enabled

Lending: Not Enabled

Enhanced Typesetting: Not Enabled

Best Sellers Rank: #116,875 Paid in Kindle Store (See Top 100 Paid in Kindle Store) #31

inÂ Books > Computers & Technology > Programming > Languages & Tools > C & C++ > Tutorials #61 inÂ Kindle Store > Kindle eBooks > Computers & Technology > Programming > C & C++ #68 inÂ Kindle Store > Kindle eBooks > Computers & Technology > Programming > Software Design > Software Development

Customer Reviews

While I was working on my PhD in computer science, part of my job as a TA was to run the computer lab. I never cared for Java, but I learned enough of it to be able to help the undergraduates when they got stuck working on their homework assignments.What struck me is that often the undergrads know more Java than I did; their problem was that they didn't understand how to solve problems. Once I walked them through the process of designing a solution, then they could write the program. When I interviewed at Microsoft, the interviewer said the same thing: that many of the people he had talked to were not able to even answer the first interview question (which required figuring out a solution to a problem and then coding it).As such, it's no surprise that I was happy to see this book, with its promise of helping people understand how to solve problems rather than simply how to write code. The first chapter immediately dives in to solving some logic puzzles; while these aren't computer related (and some are classic problems that everyone knows) they get the point across that programming is about solving problems. The actual language is secondary; what's important is being able to break the problem down to the relevant information and figure out a way to solve it. Once you have an approach that allows you to tackle the problem, then you can figure out how to do each individual step.Chapter two switches to solving problems using C++, rather than generic logic puzzles, and then we're off and running. We follow that with one chapter each on solving problems using arrays, pointers and dynamic memory, classes, recursion, and code reuse. Finally we have a chapter about working to your strengths as a programmer to find solutions efficiently.

I recently read Andy Hunt and Dave Thomas' "The Pragmatic Programmer". I thought to myself that it was interesting to see this book from the perspective of 13 years later and what was still being practiced actively and where we may have moved on, as well as the suggestions they made to be effective and, yes, pragmatic programmers.To book end this experience, I wanted to look at another title that was in a similar vein, but much more recent, as in this title was released just a few months ago. This book, V. Anton Spraul's "Think Like a Programmer", covers much of the same ground as "The Pragmatic Programmer", but does so with a much narrower scope. While "The Pragmatic

Programmer" looked to focus on many different aspects of being effective, "Think Like a Programmer" puts the bulk of its energy on one issue; problem solving and the tools necessary to approach problems and develop solutions. The goal of this book is to help answer that age old challenge... I understand the syntax, I can read code, I can modify code, I can work with other people's code and understand what it's doing, but when I sit down in front of a blank editor, I'm lost!"Think Like a Programmer" takes the reader down a number of paths to help explain some of programmings more challenging aspects and do so with generally basic coding structures. The entire book's examples are in C++, so there is a unity to the problems being presented. While the examples are in C++, all but a few of the problems could be ported to other languages like Java, Ruby, Python, etc.. The Chapter on pointers might be the sole exception, and the chapter on classes comes from the perspective of a language that can be used with both procedural and object oriented approaches.

One thing I have noticed over the years is that at least 95% of programmers spend a long time writing what ought to be relatively simple code and that they usually end up with a lot more code than they really needed. The problem is that these programmers have only learnt half of what they really need to know in order to be able to write programs - the programming language itself. The other half of programming is a creative process that is not taught in most programming classes and which in part is a talent that can't be taught.This book sets out to teach the part of the creative part of programming that can be taught to those who don't have the talent of visualising solutions to complex problems. Anyone who uses the techniques taught in this book will eventually find their way into that top 5% of programmers who can write good solutions to complex programming problems. As they use creative problem solving to produce more effective code their programming ability will improve even though they may have no greater understanding of the particular programming language syntax than they did at the start.While just about everything in this book seemed trivial and obvious to me (but then I'm not part of the target audience for the book), experience has shown me that such things are seldom either trivial or obvious to most other programmers and I have not previously seen any resource that explains these processes as clearly as this book does. This is definitely a book that I would use in teaching programming to others,The author has chosen C++ as the language used for the solutions to the various problems the book uses to demonstrate the problem solving process.

Download to continue reading...

Think Like a Programmer: An Introduction to Creative Problem Solving ACT LIKE A LADY, THINK

LIKE A MAN:By Steve Harvey:Act Like a Lady, Think Like a Man: What Men Really Think About Love, Relationships, Intimacy, and Commitment [Hardcover] 1st Edition The Graphic Designer's Digital Toolkit: A Project-Based Introduction to Adobe Photoshop Creative Cloud, Illustrator Creative Cloud & InDesign Creative Cloud (Stay Current with Adobe Creative Cloud) Act Like a Lady, Think Like a Man, Expanded Edition: What Men Really Think About Love, Relationships, Intimacy, and Commitment Professional ASP.NET 2.0 Design: CSS, Themes, and Master Pages (Programmer to Programmer) Swift: Programming, Master's Handbook: A TRUE Beginner's Guide! Problem Solving, Code, Data Science, Data Structures & Algorithms (Code like a PRO in ... mining, software, software engineering,) Java Programming: Master's Handbook: A TRUE Beginner's Guide! Problem Solving, Code, Data Science, Data Structures & Algorithms (Code like a PRO in 24 ... design, tech, perl, ajax, swift, python) Ruby: Programming, Master's Handbook: A TRUE Beginner's Guide! Problem Solving, Code, Data Science, Data Structures & Algorithms (Code like a PRO in 24 ... design, tech, perl, ajax, swift, python) Think Like an Artist: How to Live a Happier, Smarter, More Creative Life Act Like a Success, Think Like a Success: Discovering Your Gift and the Way to Life's Riches How To Play Tennis: Think Like A Pro, Not Like An Amateur Java: An Introduction to Problem Solving and Programming (7th Edition) Java: An Introduction to Problem Solving and Programming (4th Edition) Java: An Introduction to Problem Solving and Programming (6th Edition) Introduction to Programming with Java: A Problem Solving Approach Java: Introduction to Problem Solving and Programming (5th Edition) Student Value Edition for Java: An Introduction to Problem Solving and Programming (6th Edition) MATLAB - Programming with MATLAB for Beginners - A Practical Introduction to Programming and Problem Solving (Matlab for Engineers, MATLAB for Scientists, Matlab Programming for Dummies) Introduction to Counting & Probability (The Art of Problem Solving) Introduction to Geometry, 2nd Edition (The Art of Problem Solving)

Dmca