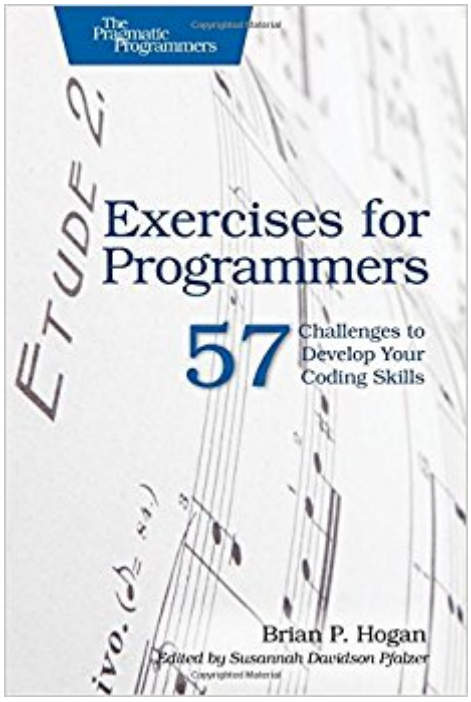
Exercises For Programmers: 57 Challenges To Develop Your Coding Skills



Synopsis

When you write software, you need to be at the top of your game. Great programmers practice to keep their skills sharp. Get sharp and stay sharp with more than fifty practice exercises rooted in real-world scenarios. If you’re a new programmer, these challenges will help you learn what you need to break into the field, and if you’re a seasoned pro, you can use these exercises to learn that hot new language for your next gig.One of the best ways to learn a programming language is to use it to solve problems. That’s what this book is all about. Instead of questions rooted in theory, this book presents problems you’ll encounter in everyday software development. These problems are designed for people learning their first programming language, and they also provide a learning path for experienced developers to learn a new language quickly.Start with simple input and output programs. Do some currency conversion and figure out how many months it takes to pay off a credit card. Calculate blood alcohol content and determine if it’s safe to drive. Replace words in files and filter records, and use web services to display the weather, store data, and show how many people are in space right now. At the end you’ll tackle a few larger programs that will help you bring everything together.Each problem includes constraints and challenges to push you further, but it’s up to you to come up with the solutions. And next year, when you want to learn a new programming language or style of programming (perhaps OOP vs. functional), you can work through this book again, using new approaches to solve familiar problems.What You Need:You need access to a computer, a programming language reference, and the programming language you want to use.

Book Information

Paperback: 118 pages
Publisher: Pragmatic Bookshelf; 1 edition (September 14, 2015)
Language: English
ISBN-10: 1680501224
ISBN-13: 978-1680501223
Product Dimensions: 6 x 0.2 x 9 inches
Shipping Weight: 12.6 ounces (View shipping rates and policies)
Average Customer Review: 4.0 out of 5 stars 19 customer reviews
Best Sellers Rank: #161,454 in Books (See Top 100 in Books) #48 inÃ Â Books > Textbooks > Computer Science > Algorithms #114 inÃ Â Books > Computers & Technology > Programming > Algorithms #198 inÃ Â Books > Textbooks > Computer Science > Software Design & Engineering

Customer Reviews

View larger Q&A with Brian Hogan, author of Exercises for Programmers Why did you decide to write this book? I learned to program when I was in fourth grade. I was struggling with some math problems at the time, and my dad showed me how to write a program to quiz me at math problems. My dad wasn't formally trained, he just knew enough to show me what to do. And so programming, to me, was about solving problems. But when I got to college, my professors were more interested in doing algorithms and puzzles. I was never great at mentally connecting the dots. But I had one teacher who was very focused on real-world programming; writing programs to solve business problems. And everything clicked. When I got into the field, I found myself in many situations where I was teaching people to code, and I needed exercises for them to do, so I started looking at the things I had to write at work and simplifying them down. I've written BMI calculators, widgets for web sites that pulled down the weather, URL shorteners, and many other things that can teach programming concepts in context. So when I became a teacher full-time a few years ago, I began introducing these exercises into my classes for additional practice in order to prepare students for assessments. I saw student performance improve significantly. And I figured that if it worked for me, it would work for everyone. So this book is for beginners? Over the years I've had to learn some new programming languages, and I've returned to these programs to get me through that. When I was learning Go a few years ago, I tried these programs. And I just did the same thing this last year with Elixir. I've seen how the "todo list" program has become the way for developers to get their minds around an MVC framework, so I think there's a ton of value in solving known problems with a new language. I also think it's easier to learn a language when you have some goals and direction. When you've never used Swift before, even something as simple as making a mad-lib program can be a great experience. What's your favorite exercise in this book? One of the exercises in the book uses an API to show you how many people are in space. The API shows you their names and which spacecraft they are on. First, I think it's awesome we live in a time where people are in space. But also, I think the exercise is interesting and engaging, while still having you work through the concepts of pulling down remote data and formatting it.

View larger What do you hope readers take away from the book? I think we get better with practice. If you are playing piano and you only go to your lessons, and you never practice in between, you won't be as good as you could be. And I think that is the same with writing code. I think if you go through a degree program and only do the work that's assigned, you won't get as much experience. And I believe that the more languages you explore, the better

you'll get at solving problems. So I hope that by reading this book, people will be inspired to practice with the language they know, or even to try a new language.

Brian Hogan is a developer, author, and teacher who loves building things for the web. He teaches introductory programming classes at the college level and has an interest in performance-based learning. He is the author of Automate with Grunt, tmux and HTML5 and CSS3 and is the co-author of Web Development Recipes.

This is a great book to get you thinking about how to solve problems like a programmer and for learning the nuances of the language you're working in. I am a seasoned iOS programmer and I purchased this book to use as a supplement to reading the Apple materials on Swift. Using this book with Swift is semi-challenging and it forces you to think very definitely than you would programming with Obj-C.I recommend this book. It's pricey for what it is length-wise but still worth it.

This is a set of tasks to create small to medium size programs, that aims at providing programmers with problems to practice their skills. The book is divided in several topics, and begins with very easy problems which get progressively harder. Each task has several additional challenges, to make the problem more complicated. The main focus is to master the reader's chosen programming language/API. And as such the target audience of the book is people with basic to intermediate knowledge in the language they use with this book, because they need to delve deeply into the language manual/specification.

It's a good book. I want to develop my skills in any programming language , This book is going to help to improve those skills in programming as the author of this book says that it will help to develop those skills in any programming language. i really recommend if you want to test yourself this is the right book

I would say that the exercises in this material are best suited to intermediate-level programmers. If you have ever had a job that involved writing code, chances are this is likely to be too easy.This book would be excellent for a freshman CS student to work on in or out of the classroom, as a bit of muscle memory for later coding practices, but I would not recommend it for anyone who is looking to improve already passable coding.

Did many of these things in a course in high school. Its a great refresher and reminder of the different commands I used, and didn’t use. Great example problems that test your skills!

An inspiring little book

Went through pretty much all the exercises while learning Python during Hurricane Matthew. This is fantastic for both beginners and seasoned coders (like myself) to get some exercise in new programming languages.

The description on the back cover of this book claims that this book will "help you get to the next level of expertise, whether you’re a new programmer or a seasoned pro". I beg to differ.For your convenience, here’s a brief breakdown by chapter titles:1. Turning Problems into Code (brief intro, no exercises)2. Input, Processing, and Output (6 exercises)3. Calculations (7 exercises)4. Making Decisions (mostly about control statements, 10 exercises)5. Functions (4 exercises)6. Repetition (loops, 5 exercises)7. Data Structures (arrays/lists/maps, 8 exercises)8. Working with Files (file i/o, 6 exercises)9. Working with External Services (interacting with web services, 6 exercises)10. Full Programs (5 exercises)The vast majority of the exercises here are dedicated to topics that should be covered by any "introduction to programming" class or free online language tutorials. The exercises ramp up on difficulty at an absurdly slow pace - for example, the classic "Guess the Number" game doesn’t make an appearance until more than halfway through the book. And in the spirits of "practice makes permanent", the book devotes TEN exercises on just writing if-else statements. Seriously?!Even the later exercises lack variety. Most of the exercises in the "Full Programs" chapter are simple applications that deal with some external data store. Instead of a taking a broad look at all the various problems that can be solved through programming, the book just stays within a very limited comfort zone.Plenty of topics were omitted in this book, but really should have been included. In fact, these topics are what truly demonstrates the differences between programming languages and their design philosophies. For example:- Inheritance vs composition (ex: Java and Python are both OO languages but handle this completely differently)- Multi-threading and concurrence (ex: Ruby and Go have very elegant syntax for this)- Common design patterns (ex: implement a singleton in the language of your choice)- Interprocess Communication (ex: pipes in C/C++)- More advanced algorithms and data structures (ex: implement BFS or DFS) ...and the list goes on.In short, only get this book if you are just learning to write code, or have been out of practice for too long and need a refresher. Experienced engineers looking for a

challenge should look elsewhere.

Exercises for Programmers: 57 Challenges to Develop Your Coding Skills Memory Exercises: Memory Exercises Unleashed: Top 12 Memory Exercises To Remember Work And Life In 24 Hours With The Definitive Memory Exercises Guide! (memory exercises, memory, brain training) Coding in the Real World (Kids Get Coding) (Kids Get Coding (Paper)) How to Draw and Paint Portraits: Learn how to draw people through taught example, with more than 400 superb photographs and practical exercises, each designed to help you develop your skills How to Purchase and Develop Commercial Real Estate: A Step by Step Guide for Success (How to Develop Commercial Real Estate Book 1) Creating Glass Beads: A New Workshop to Expand Your Beginner Skills and Develop Your Artistic Voice Language Implementation Patterns: Create Your Own Domain-Specific and General Programming Languages (Pragmatic Programmers) Provider's Coding Notes: Billing & Coding Pocket Guide (Davis's Notes) ICD-10-CM and ICD-10-PCS Coding Handbook, 2014 ed., with Answers (ICD-10- CM Coding Handbook W/Answers) Coding Notes: Pocket Coach for Medical Coding Conquer Medical Coding 2016: A Critical Thinking Approach with Coding Simulations ICD-10-CM 2017 Snapshot Coding Card: Dermatology (ICD-10-CM 2017 Snapshot Coding Cards) ICD-10-CM 2017 Snapshot Coding Card: Cardiology (ICD-10-CM 2017 Snapshot Coding Cards) ICD-10-CM 2017 Snapshot Coding Card: Obstetrics / Gynecology (ICD-10-CM 2017 Snapshot Coding Cards) ICD-10-CM 2017 Snapshot Coding Card: Emergency Medicine (ICD-10-CM 2017 Snapshot Coding Cards) ICD-10 Snapshot 2017 Coding Cards Psychiatry (ICD-10-CM 2017 Snapshot Coding Cards) ICD-10-CM 2017 Snapshot Coding Card: Ophthalmology (ICD-10-CM 2017 Snapshot Coding Cards) Understanding Coding for the Non-Coder: The Relationship Between Coding, Payment and Documentation and Their Impact on Health Care E&M Coding Clear & Simple: Evaluation & Management Coding Worktext CPT 2017 Express Reference Coding Card: CPT & HCPCS Modifiers (CPT 2017 Express Reference Coding Cards)