	Book
16	Code Complete: A Practical Handbook of Software Construction
15	The Pragmatic Programmer: From Journeyman to Master
14	The Mythical Man-Month
13	Refactoring: Improving the Design of Existing Code
12	Clean Code: A Handbook of Agile Software Craftsmanship
11	Introduction to Algorithms
10	Patterns of Enterprise Application Architecture
8	Peopleware: Productive Projects and Teams
8	Working Effectively with Legacy Code
8	Design Patterns: Elements of Reusable Object-Oriented Software
7	Code: The Hidden Language of Computer Hardware and Software
7	Domain-Driven Design: Tackling Complexity in the Heart of Software
6	Head First Design Patterns
6	Structure and Interpretation of Computer Programs
6	Programming Pearls
6	Coders at Work: Reflections on the Craft of Programming
5	The C Programming Language
5	The Art of Computer Programming
5	and Deployment Automation
4	Rapid Development: Taming Wild Software Schedules
4	Agile Software Development, Principles, Patterns, and Practices
4	The Design of Everyday Things: Revised and Expanded Edition
4	Usability
4	Effective Java
4	Gödel, Escher, Bach: An Eternal Golden Braid
4	The Clean Coder: A Code of Conduct for Professional Programmers
3	User Stories Applied: For Agile Software Development
3	Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions
3	Release It!: Design and Deploy Production-Ready Software

3 The Passionate Programmer: Creating a Remarkable Career in Software Development

2 About Face: The Essentials of Interaction Design

2 Designing Web Usabillity

2 A Pattern Language: Towns, Buildings, Construction (Center for Environmental Structure)

2 The Visual Display of Quantitative Information

2 Extreme Programming Explained: Embrace Change

2 Smalltalk Best Practice Patterns

2 Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives

2 Refactoring to Patterns

2 Agile Estimating and Planning

2 Zen and the Art of Motorcycle Maintenance: An Inquiry Into Values

2 Cracking the Coding Interview: 189 Programming Questions and Solutions

2 Apprenticeship Patterns: Guidance for the Aspiring Software Craftsman

2 Pragmatic Thinking and Learning: Refactor Your Wetware

2 The Art of Agile Development

2 Soft Skills: The software developer's life manual

2 Succeeding with Agile: Software Development Using Scrum

2 Growing Object-Oriented Software, Guided by Tests

2 Java Concurrency in Practice

2 Software Estimation: Demystifying the Black Art (Developer Best Practices)

2 Effective Modern C++: 42 Specific Ways to Improve Your Use of C++11 and C++14

2 Algorithms

2 Seven Languages in Seven Weeks: A Pragmatic Guide to Learning Programming Languages (Pragmatic Programmers)

2 Debugging: The 9 Indispensable Rules for Finding Even the Most Elusive Software and Hardware Problems

2 Rework

1 A Random Walk down Wall Street: The Time-tested Strategy for Successful Investing
1 Agile Testing: A Practical Guide for Testers and Agile Teams
1 Beautiful Evidence
1 Designing Interfaces
1 Envisioning Information
1 Essential Scrum: A Practical Guide to the Most Popular Agile Process
1 Foundations of Security: What Every Programmer Needs to Know

1 Growing a Business

1 Helplessness: On Depression, Development, and Death

1 How to Win Friends and Influence People

1 Influence: The Psychology of Persuasion
1 Leading Lean Software Development: Results Are not the Point

1 Microserfs

1 Object Design: Roles, Responsibilities, and Collaborations

1 Regular Expressions Cookbook

The Inmates Are Running the Asylum: Why High Tech Products Drive
1 Us Crazy and How to Restore the Sanity
The Lean Startup: How Today's Entrepreneurs Use Continuous
1 Innovation to Create Radically Successful Businesses

1 The Non-Designer's Design Book

1 User Interface Design for Programmers
1 Visual Explanations: Images and Quantities, Evidence and Narrative

1 Troubled IT Projects : Prevention and Turnaround

Collaboration Explained: Facilitation Skills for Software Project
1 Leaders

1 Agile Software Development: The Cooperative Game

1 The Wisdom of Teams: Creating the High-Performance Organization

Inside the Machine: An Illustrated Introduction to Microprocessors
1 and Computer Architecture
1 The Algorithm Design Manual
1 Hot Text: Web Writing that Works
1 Notes on the Synthesis of Form
1 The Process of Creating Life: Nature of Order
The Innovator's Dilemma: The Revolutionary Book That Will Change
1 the Way You Do Business

1 Are Your Lights On?: How to Figure Out What the Problem Really Is
1 The Elements of Style
1 On Writing Well: The Classic Guide to Writing Nonfiction

1 Thinking in Java

1 Testing Computer Software

1 Ship it! A Practical Guide to Successful Software Projects

1 How to Win Friends & Influence People

1 The War of Art: Break Through the Blocks and Win Your Inner Creative Battles

1 Agile!: The Good, the Hype and the Ugly

1 Build an Awesome PC, 2014 Edition: Easy Steps to Construct the Machine You Need

1 The CERT C Coding Standard, Second Edition: 98 Rules for Developing Safe, Reliable, and Secure Systems

1 When Computing Got Personal: A history of the desktop computer

1 Penetration Testing: A Hands-On Introduction to Hacking

1 Pattern Hatching: Design Patterns Applied

1 Beyond Software Architecture: Creating and Sustaining Winning Solutions

1 Agile Project Management with Scrum (Developer Best Practices)

1 Agile Project Management: Creating Innovative Products

1 Compilers: Principles, Techniques, and Tools

1 Zero Bugs and Program Faster

1 xUnit Test Patterns: Refactoring Test Code

1 Land the Tech Job You Love (Pragmatic Life)

1 Presentation Patterns: Techniques for Crafting Better Presentations

1 Test Driven Development: By Example

1 Programming with POSIX Threads

1 Pattern-Oriented Software Architecture Volume 2: Patterns for Concurrent and Networked Objects

1 Object Thinking (Developer Reference)

1 PMP Exam Prep

1 The Art of Software Testing

1 XML in a Nutshell

1 Writing Effective Use Cases

1 Software Requirements

1 Version Control with Git: Powerful tools and techniques for collaborative software development

1 JavaScript: The Definitive Guide: Activate Your Web Pages (Definitive Guides)

1 CSS: The Definitive Guide

1 Crystal Clear: A Human-Powered Methodology for Small Teams: A Human-Powered Methodology for Small Teams

1 Agile Documentation : A Pattern Guide to Producing Lightweight Documents for Software Projects

1 After the Gold Rush: Tarnished Dreams in the Sacramento Valley (Revisiting Rural America)

1 Software Craftsmanship: The New Imperative

1 UML Distilled: A Brief Guide to the Standard Object Modeling Language

1 Understanding the Professional Programmer

1 JavaScript: The Good Parts

1 Beautiful Code: Leading Programmers Explain How They Think
Seven Databases in Seven Weeks: A Guide to Modern Databases
1 and the NoSQL Movement

1 Seven Concurrency Models in Seven Weeks: When Threads Unravel

1 The New Turing Omnibus: Sixty-Six Excursions in Computer Science
How to Design Programs: An Introduction to Programming and
1 Computing

1 How to Prove It: A Structured Approach
The Annotated Turing: A Guided Tour Through Alan Turing's Historic
1 Paper on Computability and the Turing Machine

1 Introduction to Graph Theory
Paradigms of Artificial Intelligence Programming: Case Studies in
1 Common Lisp
Artificial Intelligence for Humans, Volume 3: Deep Learning and
1 Neural Networks
AntiPatterns: Refactoring Software, Architectures, and Projects in
1 Crisis

1 Test Driven Development: By Example
The Phoenix Project: A Novel About IT, DevOps, and Helping Your
1 Business Win
The Cathedral & the Bazaar: Musings on Linux and Open Source by
1 an Accidental Revolutionary

1 Practical Object-Oriented Design in Ruby: An Agile Primer

Recommendations

engineer.

they've discovered during their respective careers as designers of software and writers of

the book still is to software product development. If you are involved in software, this book is

to use the refactoring, step by step instructions for implementing it, and an example

clean code.

doing a lot of in software development).

problems an architect or a developer runs into when working on a typical software

sections: Managing humans, managing the work environment, finding the right people,

to support and work with a legacy system.

regularly refer to it when I'm stuck with a piece of design work. Highly recommended.

explain about the operating system, floating point arithmetic and GUIs.

applicable and I have been successful in eliminating anemic domain models and handling

oriented software and the book is a pleasure to read. Highly recommend!

this book is oriented toward engineering.

teach something new but to help you become a better problem solver.

introduction in the format of transcripts.

programmer, and you've got a terrific foundation for everything else you're going to learn.

particularly on information structures–the representation of information inside a computer, the

process—and trust me, you do!

words, "read the instructions on the paint can before painting." Sure, it sounds obvious

only that but it introduces the SOLID design principles which are fundamental to

the no-brainer we thought it was, maybe it's time to give ourselves a break for not being able

information, and it's presented in a concise, approachable format. It's suitable for any

Many of the principles are also applicable to other programming languages, as a lot of the principles of good software design are universal ones. I found that, after reading this book, I looked at the code that I wrote in a different light - it became easier to read, there were less lines of code overall, and it was easier to fix bugs when I came back into it later.

https://www.infoq.com/articles/recommended_reading_list

My primary criterion for choosing courses was to cover topics that were raised in this magnificant, panoramic, and brilliantly interesting book: AI, cognitive science, computer science, philosophy, psychology, music, and art are woven together magically.

http://www.joelonsoftware.com/navlinks/fog0000000262.html

A lot of developers won't like this programming book, because the advice is sometimes difficult to swallow. But swallow it if you can, because Bob Martin, the author of the book, is a veteran in the industry and has managed to stay relevant decade after decade of pumping out code. I really found this book helpful in shaping my career and making tough decisions.

https://simpleprogrammer.com/2015/03/23/the-ultimate-list-of-programming-books/

Another great Agile book that helps with a major trouble area: user stories. I've worked with some pretty crappy user stories when working on Agile teams, simply because no one knew how to make good ones or what user stories were actually supposed to look like. This book solves that problem.

https://simpleprogrammer.com/2015/03/23/the-ultimate-list-of-programming-books/

This has been one of my favorite books since 2004 when I first read the book. The authors have nicely organized and described several asynchronous design patterns useful in every day enterprise applications, especially those involving multi-system integration.

https://www.infoq.com/articles/recommended_reading_list

The book is filled with war stories showing where large-scale systems fail, and Michael uses his vast experience to extract some common anti-patterns, problems that occur again and again in naive designs.

https://www.infoq.com/articles/recommended_reading_list

I highly recommend reading this book if you want to move beyond programming as just a job.
https://simpleprogrammer.com/2015/03/23/the-ultimate-list-of-programming-books/
I've owned a few versions of this book now (this is version four), and it is the rare book which is getting better and better as it is revised, and more authors are added for different perspectives.
https://blog.codinghorror.com/recommended-reading-for-developers/

Designing Web Usability is of course a full-on web usability primer, so it's a bit different than the GUI-oriented Cooper books.
https://blog.codinghorror.com/recommended-reading-for-developers/
I bought it because I'm interested in architecture. Then I noticed something: almost everything in the book can be applied to the work we do as software designers.
http://www.joelonsoftware.com/navlinks/fog0000000262.html
https://blog.codinghorror.com/recommended-reading-for-developers/
Extreme Programming Explained (Kent Beck) is a good intro to XP, and even if you don't subscribe to the everything-agile approach, it makes a good read. Probably not a reference tome though.
https://www.infoq.com/articles/recommended_reading_list
Incredibly practical advice for what constitutes good OO code. It’s done in Smalltalk, but the principles are mostly universal. Probably my favorite nuts’n’bolts of programming design book. Very granular.
https://signalvnoise.com/posts/3375-the-five-programming-books-that-meant-most-to-me
This software architecture book is a must read for beginners. This book focuses on many common mistakes people make during beginning of the software architect job. This book can certainly help you do your job more efficiently.
http://www.fromdev.com/2010/08/best-software-architecture-books-must.html
This is a programming book that teaches you—scratch that—shows you how to move existing code into patterns and how to even move it out of patterns. Should be on every architect’s bookshelf.
https://simpleprogrammer.com/2015/03/23/the-ultimate-list-of-programming-books/

Highly recommended for anyone working in an Agile environment.
https://simpleprogrammer.com/2015/03/23/the-ultimate-list-of-programming-books/
This book goes a long way towards relating engineering and philosophy.
http://www.joelonsoftware.com/navlinks/fog0000000262.html
The book is full of great interview advice and real programming problems that will not only help you pass a coding interview, but make you a better programmer overall. If you can master the exercises in this book, it will be very difficult to stump you in a programming interview.
https://simpleprogrammer.com/2015/03/23/the-ultimate-list-of-programming-books/
Apprenticeship Patterns is the best book on Software Craftsmanship I've read, and I've read quite a few.
http://www.nomachetejuggling.com/2014/02/05/top-10-career-changing-programming-books/
It contains many ways to help improve learning, working etc., for one by using the Dreyfus learning model.
https://www.infoq.com/articles/recommended_reading_list

Each article in the book explains a practice or a component of Agile Development. Thus, the book provides practical value to businesses, customers, testers, analysts and developers.
http://www.fromdev.com/2012/02/agile-development-books-must-read.html
This book covers everything-else-apart-from-coding ranging from career, to personal branding, blogging, learning, teaching, finances, and even fitness and relationships.
http://www.codingdojo.com/blog/9-best-programming-books-read-right-now-want-distinguish/

It provides an insight into Agile Development and compares some of the real ideas with stories, also offers examples of Agile Development which will appeal to a number of readers. Moreover, it provides tips to adopt new improved Agile Development procedures.
http://www.fromdev.com/2012/02/agile-development-books-must-read.html
You definitely should read it, just to know what's going on, but it won't help you write good tests. Read this one instead, and read it many times.
http://www.yegor256.com/2015/04/22/favorite-software-books.html

This book is critical. You shouldn't let even senior developers touch your code base if they haven't read this book and understood the concepts within. It is old, but it's still relevant.
https://blog.parasoft.com/software-development-books-to-read
This one's an interesting read about software engineering and its most tricky part—estimations. At the least, read it to be aware of the problem and possible solutions.
http://www.yegor256.com/2015/04/22/favorite-software-books.html
This books are totally crucial for any serious C++ developer out there. They cover a lot of useful techniques about improving your C++ skills and understanding the true power that the language offers.
https://nicolasgoles.com/blog/2011/02/10-favorite-software-development-books/

https://www.reddit.com/r/learnprogramming/wiki/books
I love this book, because it stretches you and makes you a more open-minded programmer. This book helped me to see how similar so many programming languages are, appreciate their differences and see just how fast I could learn.
https://simpleprogrammer.com/2015/03/23/the-ultimate-list-of-programming-books/
All new developers on my team are asked to read this book. This book is a quick and fun read with interesting examples that anyone can understand.
https://blog.parasoft.com/software-development-books-to-read
This book is for project leads and managers who want to learn from what has made others successful in writing software. You don't have to agree with everything they have to share in order to come away with new thoughts and ideas.
https://blog.parasoft.com/software-development-books-to-read
What if I told you that you could read one book and know everything you are going to need to know about managing your investments? And I mean, everything. Well, it's true. And this is the book.
http://www.joelonsoftware.com/navlinks/fog0000000262.html
https://stevewedig.com/2014/02/03/software-developers-reading-list/
https://blog.codinghorror.com/recommended-reading-for-developers/
https://stevewedig.com/2014/02/03/software-developers-reading-list/
https://blog.codinghorror.com/recommended-reading-for-developers/
https://stevewedig.com/2014/02/03/software-developers-reading-list/
https://stevewedig.com/2014/02/03/software-developers-reading-list/
People regularly email me and say, "gosh, I love your theory about starting a company the Ben and Jerry's way, but, how do I get started?" This is the book you want to read.
http://www.joelonsoftware.com/navlinks/fog0000000262.html
What does this have to do with a book on depression? Well, it turns out that people literally become clinically depressed when they feel like they can't control their lives and their environment.
http://www.joelonsoftware.com/navlinks/fog0000000262.html
Your success relies on other people's cooperation, collaboration, and ideas more than ever before. How to Win Friends and Influence People will help you navigate these personal waters better than any other resource I know of.
http://robertgreiner.com/2013/09/software-developer-book-recommendations/
Cialdini's excellent book discusses the psychological theories behind the science and practice of influencing the behavior of other people. Read it before they do!
http://www.joelonsoftware.com/navlinks/fog0000000262.html
https://stevewedig.com/2014/02/03/software-developers-reading-list/
Nothing quite captures the feeling of being a young programmer at a big software company as well as Microserfs.
http://www.joelonsoftware.com/navlinks/fog0000000262.html

https://stevewedig.com/2014/02/03/software-developers-reading-list/
Once you delve into the world of regular expressions, you may become drunk with the amazing power and potential they have, which results in things like Perl. Remember, absolute power corrupts absolutely. But it also rocks absolutely.
https://blog.codinghorror.com/recommended-reading-for-developers/
There's nothing magical here; as always, it boils down to knowing who your users are and what they really do – and the personas technique is a great way to get there.
https://blog.codinghorror.com/recommended-reading-for-developers/

https://stevewedig.com/2014/02/03/software-developers-reading-list/
This excellent, thin book will give you a grasp of the principles behind page layout, fonts, etc. The good news is, you can read it in the bath before the water gets cold, and the next day, your dialog boxes and powerpoints and web pages will start looking better.
http://www.joelonsoftware.com/navlinks/fog0000000262.html
The most common reaction I've heard from readers is "after reading your book, I found three things I just HAVE to change in my program."
http://www.joelonsoftware.com/navlinks/fog0000000262.html
https://blog.codinghorror.com/recommended-reading-for-developers/
...probably one you've not come across before, but actually explains a number of things that can go wrong in managing a project.
https://www.infoq.com/articles/recommended_reading_list
If you hate meetings and/or believe they should improve, you must read this book - whether you are involved in an agile project or not!
https://www.infoq.com/articles/recommended_reading_list
Possibly the most interesting book I've ever read about agile software development.
https://www.infoq.com/articles/recommended_reading_list
By the end of the book you will discover that "Nothing can guarantee the creation of high performance teams. The best you can do is put in place the conditions that will help them form."
https://www.infoq.com/articles/recommended_reading_list
nowing how instruction fusion helps improve reorder buffer efficiency is great for water cooler discussions - but knowing how memory hierarchy and cache behavior impacts performance on modern CPUs might just help you optimize yourself out of a tight corner.
https://www.infoq.com/articles/recommended_reading_list
https://www.reddit.com/r/learnprogramming/wiki/books
http://feltpresence.com/reading_list
http://feltpresence.com/reading_list
http://feltpresence.com/reading_list

http://feltpresence.com/reading_list
This isn’t technically a programming book, but it deals with the biggest problem facing developers none the less. Nothing has increased my programming productivity more than being able to restate hard problems as simple ones.
https://signalvnoise.com/posts/3375-the-five-programming-books-that-meant-most-to-me
https://signalvnoise.com/posts/3375-the-five-programming-books-that-meant-most-to-me
https://signalvnoise.com/posts/3375-the-five-programming-books-that-meant-most-to-me
I don’t think I’ve ever encountered a better book on learning a programming language than this book. Even if you have no interest in Java, I recommend reading it.
https://simpleprogrammer.com/2015/03/23/the-ultimate-list-of-programming-books/
After reading this book, I was able to communicate with QA better, write more testable code and avoid defects by learning how to test my own code before throwing it over the wall.
https://simpleprogrammer.com/2015/03/23/the-ultimate-list-of-programming-books/
It’s a great book to read and then to give to your manager to highlight the value of some best practices like: continuous integration, automated testing, scaled back planning, etc.
https://simpleprogrammer.com/2015/03/23/the-ultimate-list-of-programming-books/
As a software developer, you will deal with people during your entire career. If you learn how to deal with them effectively, you’ll have a much better go of it. So, I highly recommend reading this book. This book changed my life and set me on the path I am on now.
https://simpleprogrammer.com/2015/03/23/the-ultimate-list-of-programming-books/

This book is the reason why I can sit down and spend 4 hours writing this blog post. It’s an excellent book that will inspire you to be your best, and finally beat procrastination.
https://simpleprogrammer.com/2015/03/23/the-ultimate-list-of-programming-books/
In the field of applied Agile, this is by a wide margin the most insightful, thought-provoking, well-written book.
http://www.drdobbs.com/cpp/developer-reading-list/240168591
For those who actively (or secretly) hanker to build their own systems, this book is definitely the right guide — a great help, and a lot of fun.
http://www.drdobbs.com/cpp/developer-reading-list/240168591
Few developers, especially in regulated industries, would disagree that programmers need to learn to code automatically using secure techniques and hardened functions — most especially in C. This volume, which at 512 pages provides all the necessary detail, is bound to get them across the river Jordan to that happy, secure place. Definitely recommended.
http://www.drdobbs.com/cpp/developer-reading-list/240168591?pgno=3
For readers who were programming in that era or who, like me, caught the tail end of it, there will be the rediscovery of lived history with the unique clarity that research can bring to memories made less sharp by time. The book will also fill in history and context that might not have been obvious.
http://www.drdobbs.com/cpp/developer-reading-list/240168591?pgno=5
The book is clear, easy-to-read, and filled with numerous illustrations. Many processes are explained via step-by-step directions. While it might breed a new generation of script kiddies, it's likely to also breed software developers who give such attackers much less to exploit.
http://www.drdobbs.com/cpp/developer-reading-list/240168591?pgno=6
Chapter 2 of this book is a must read where John develops a file system application. This book talks in great detail about which patterns suit best the needs of the application and which do not.
http://www.fromdev.com/2010/06/5-best-design-pattern-books-you-must.html
This book delivers on its promise to discuss the larger business realities of creating software products. If you're a software architect, or dream of being one, this is a must read book.
http://www.fromdev.com/2010/08/best-software-architecture-books-must.html
This is the best agile book out of the lot. The book provides a clear explanation about Scrum, starting from the basic tools and methodologies and it helps to provide accurate methods to fulfill customer requirements and for the translation of requirements into the software functions.
http://www.fromdev.com/2012/02/agile-development-books-must-read.html
The book provides an overview on Agile Project Management and it also compares the Agile Development methodologies to traditional methods of Development.
http://www.fromdev.com/2012/02/agile-development-books-must-read.html
The 1,000-page "dragon book" focuses on compilers, but in so doing covers topics every developer should understand.
http://www.infoworld.com/article/2620356/development-tools/10-must-read-books-for-developers.html
The author of Zero Bugs spent two years researching every bug avoidance technique she could find. This book contains the best of them!
http://www.codingdojo.com/blog/9-best-programming-books-read-right-now-want-distinguish/
The book covers everything a programmer needs to become a unit testing badass, how to work with mocks and stubs, how to recognize problem smells in tests, how to refactor tests, and tons more.
http://www.nomachetejuggling.com/2014/02/05/top-10-career-changing-programming-books/
Land the Tech Job You Love is more about the mechanics of this process, how to write a resume, how to interview, how to negotiate a salary, and the like.
http://www.nomachetejuggling.com/2014/02/05/top-10-career-changing-programming-books/
This book changed my life
http://www.nomachetejuggling.com/2014/02/05/top-10-career-changing-programming-books/
Test Driven Development by the inventor of the practice...who better to learn from? The book is short, easy to understand, and presents very helpful ideas on the topic. It's very good for anyone who cares about agile software development and code quality.
https://blog.parasoft.com/software-development-books-to-read

Thread programming is fraught with obstacles and problems, most of which are non-obvious. David Butenhof takes you "down the rabbit hole", with amusing and entertaining examples, and metaphors leaving the reader with a thorough understanding of thread programming.
https://blog.parasoft.com/software-development-books-to-read
This book is to networked and concurrent objects what the "Gang of Four" is to design and structure. The patterns found herein are absolutely essential in a new and ever increasingly networked and concurrent world. Following these patterns just keeps you out of trouble in these domains.
https://blog.parasoft.com/software-development-books-to-read
This is the best book I've read about object-oriented programming, and it totally changed my understanding of it.
http://www.yegor256.com/2015/04/22/favorite-software-books.html
This book is my favorite for project management. Even though it's about the PMI approach and PMBOK in particular, it is a must-read for everyone who is interested in management.
http://www.yegor256.com/2015/04/22/favorite-software-books.html
No matter what your job description is, if you're working in the software industry, you should understand testing and its fundamental principles. This is the only book you need in order to get that understanding.
http://www.yegor256.com/2015/04/22/favorite-software-books.html
Just this one book changed everything, and ever since reading it, I've used XML everywhere. It is very well written and easy to read. It's a must for everybody.
http://www.yegor256.com/2015/04/22/favorite-software-books.html
An old and very good book, you won't actually use anything from this in your real projects, but you will pick up the philosophy of use cases, which will redirect your mind in the right direction.
http://www.yegor256.com/2015/04/22/favorite-software-books.html
A superb book about requirements analysis, the first and most important activity in any software project.
http://www.yegor256.com/2015/04/22/favorite-software-books.html
Read it from cover to cover and you will save many hours of your time in the future.
http://www.yegor256.com/2015/04/22/favorite-software-books.html
Don't read it as a practical guide (even though it's called a guide) but rather as food for thought. JavaScript offers a lot to learn for Java/Ruby/Python developers.
http://www.yegor256.com/2015/04/22/favorite-software-books.html
Every developer must know it, whether you're working with a back-end, front-end, or desktop application in C++.
http://www.yegor256.com/2015/04/22/favorite-software-books.html
Cockburn's tolerance for extensibility and flexibility offers practitioners a foot-hold for transitioning toward agile practices.
http://www.developerdotstar.com/mag/categories/software_development_book.html
Useful documentation has its place, but it should be succinct, worded simply, and presented well.
http://www.developerdotstar.com/mag/categories/software_development_book.html
McConnell posits Engineering as the ideal model for that professional maturation process, and he makes many solid arguments in its favor. Most of these points are based on the central idea of reusability
http://www.developerdotstar.com/mag/categories/software_development_book.html
Software Craftsmanship presents an alternative to the formal software engineering discipline for more typical development projects, and is well worth reading.
http://www.developerdotstar.com/mag/categories/software_development_book.html
If this acronym is not on your resume, then that is a hole you might want to fill.
http://www.developerdotstar.com/mag/categories/software_development_book.html
This is probably my favorite book on the subject of software development. It is definitely my favorite of Weinberg's many excellent books. I return to this book again and again, and find something new every time.
http://www.developerdotstar.com/mag/categories/software_development_book.html
This book is aimed at programmers with an intermediate experience who wants to know the best way to use the language.
http://www.codepancake.com/10-books-every-programmer-should-read/

https://medium.com/javascript-scene/the-software-developer-s-library-a-treasure-trove-of-books-for-people-who-love-code-f9bc92c7883b#.4qktkub59
https://medium.com/javascript-scene/the-software-developer-s-library-a-treasure-trove-of-books-for-people-who-love-code-f9bc92c7883b#.4qktkub59
https://medium.com/javascript-scene/the-software-developer-s-library-a-treasure-trove-of-books-for-people-who-love-code-f9bc92c7883b#.4qktkub59
https://medium.com/javascript-scene/the-software-developer-s-library-a-treasure-trove-of-books-for-people-who-love-code-f9bc92c7883b#.4qktkub59
https://medium.com/javascript-scene/the-software-developer-s-library-a-treasure-trove-of-books-for-people-who-love-code-f9bc92c7883b#.4qktkub59
https://medium.com/javascript-scene/the-software-developer-s-library-a-treasure-trove-of-books-for-people-who-love-code-f9bc92c7883b#.4qktkub59
https://medium.com/javascript-scene/the-software-developer-s-library-a-treasure-trove-of-books-for-people-who-love-code-f9bc92c7883b#.4qktkub59
https://medium.com/javascript-scene/the-software-developer-s-library-a-treasure-trove-of-books-for-people-who-love-code-f9bc92c7883b#.4qktkub59
https://medium.com/javascript-scene/the-software-developer-s-library-a-treasure-trove-of-books-for-people-who-love-code-f9bc92c7883b#.4qktkub59
https://medium.com/javascript-scene/the-software-developer-s-library-a-treasure-trove-of-books-for-people-who-love-code-f9bc92c7883b#.4qktkub59
https://medium.com/javascript-scene/the-software-developer-s-library-a-treasure-trove-of-books-for-people-who-love-code-f9bc92c7883b#.4qktkub59
https://medium.com/javascript-scene/the-software-developer-s-library-a-treasure-trove-of-books-for-people-who-love-code-f9bc92c7883b#.4qktkub59
https://medium.com/javascript-scene/the-software-developer-s-library-a-treasure-trove-of-books-for-people-who-love-code-f9bc92c7883b#.4qktkub59
https://medium.com/javascript-scene/the-software-developer-s-library-a-treasure-trove-of-books-for-people-who-love-code-f9bc92c7883b#.4qktkub59
The concepts in this book are not mindblowing, but they’re patiently explained, start slow, and build on each other perfectly.
http://khanlou.com/2016/06/resources-for-new-programmers/

recommend to your fellow developers.
that they've found to work in the real world into this one book.
https://blog.codinghorror.com/recommended-reading-for-developers/
pattern and then try to apply it on your code base (you don’t have to commit if it
should be this one.
https://www.infoq.com/articles/recommended_reading_list
invaluable resource to understanding the differences in architectures and why
name only, you need to pick up a copy of this book.
your new bible. Read it and take it to heart.
and become familiar with the most common design patterns you are likely to
either want to become programmers, or want to understand what programmers do, or
the ideas are worth it. It’s a great primer on how to turn a problem space into a
a renewed understanding. This book makes design patterns much easier to
then this isn't the book you are looking for. It is, however, rewarding, dense reading
programmer for a year or so. It is the collective wisdom of many journeyman coders
weaves them into a powerful view of how 15 of the industry's best and brightest
C programmers, which includes- the use of data types, if/else, for, printf, while,
algorithm, sorting & searching, semi-numerical algorithm, syntactic algorithms,
control all configuration, how to test integration points, how to handle branching and
figured out in their first decade or so of developing software on a large scale.
relatively simple object-oriented design concepts such as Meyer's Open/Closed
"UI design", even though it talks more about doors and and refrigerators than
mundane rules (like "don't change the colors of links").

Some of the recommendations are dated by the changes to the language, but overall
this is still a really good book.
https://simpleprogrammer.com/2015/03/23/the-ultimate-list-of-programming-books/

This is a huge book that is one of the most pleasurable books I have ever read. I didn
’t ever want to put this book down and I was extremely sad when I reached the end.
https://simpleprogrammer.com/2015/03/23/the-ultimate-list-of-programming-books/
The Clean Coder had a profound impact on me. It drastically altered how I talk to
bosses, product owners, project managers, marketers, salespeople, and other non-
programmers.
http://www.nomachetejuggling.com/2014/02/05/top-10-career-changing-
programming-books/

Every chapter contains a set of questions which are answered in the stories given
and the explanation is highly sensible and carefully explained.
http://www.fromdev.com/2012/02/agile-development-books-must-read.html
Don’t let the name of the book fool you, this book is all about message buses and all
of the patterns used to implement them correctly. If you are doing any kind of
integration between applications or services using a bus, you will absolutely love this
book.
https://simpleprogrammer.com/2015/03/23/the-ultimate-list-of-programming-books/
The patterns sections alone are worth the price of admission here, and the fact that
the book is chock full of even more useful content beyond them is kind of stunning
http://www.nomachetejuggling.com/2014/02/05/top-10-career-changing-
programming-books/

It's pretty high level, but full of extraordinarily important advice to ensure you find yourself at companies that fit you and that you fit into well.
http://www.nomachetejuggling.com/2014/02/05/top-10-career-changing-programming-books/

A classic of UI design, this is a great bible of GUI design from the inventor of Visual Basic.
http://www.joelonsoftware.com/navlinks/fog0000000262.html
If you're doing any kind of web design, you need to know the principles in this book. If you're doing non-web design, think of this is an excellent case study in usability engineering.
http://www.joelonsoftware.com/navlinks/fog0000000262.html
I bought it because I'm interested in architecture. Then I noticed something: almost everything in the book can be applied to the work we do as software designers.
http://www.joelonsoftware.com/navlinks/fog0000000262.html
http://feltpresence.com/reading_list
This programming book is an extremely prescriptive description of how to implement extreme programming, but the ideas in this book can be applied to many different kinds of Agile environments.
https://simpleprogrammer.com/2015/03/23/the-ultimate-list-of-programming-books/

http://feltpresence.com/reading_list

https://stevewedig.com/2014/02/03/software-developers-reading-list/
This book does an excellent job of bringing patterns into coding, rather than relegating them just to design discussions. This includes twenty-seven pattern-directed refactorings with real-world code examples.
http://www.fromdev.com/2010/06/5-best-design-pattern-books-you-must.html
his book is my second favorite and surely one of the best agile books. The book provides pragmatic approach to explain Agile Development, bringing the multiple planning approaches at several levels for measuring the software feature implementation, completion and acceptance.
http://www.fromdev.com/2012/02/agile-development-books-must-read.html
http://www.infoworld.com/article/2620356/development-tools/10-must-read-books-for-developers.html

Definitely one of the best programming interview books out there
http://www.codingdojo.com/blog/9-best-programming-books-read-right-now-want-distinguish/

https://stevewedig.com/2014/02/03/software-developers-reading-list/
Learn more about your brain than you ever realized you needed to know.
http://www.nomachetejuggling.com/2014/02/05/top-10-career-changing-programming-books/

The Art of Agile Development was easily the most influential book on how I like to work. This one is pretty subjective, as I'm pretty sure any good XP book would have had the same effect, but this was the one that did it for me, so I had to include it here.
http://www.nomachetejuggling.com/2014/02/05/top-10-career-changing-programming-books/
You will be a much more satisfied and happier person if you follow the suggestions in this book, not just as a programmer, but as your whole self.
https://dzone.com/articles/must-read-book-list-for-programmers

This is a classic for anyone transitioning to Agile. I initially read it to get a comprehensive overview of Agile and help our organization determine which strategies would be best suited to our environment.
https://blog.parasoft.com/software-development-books-to-read

https://stevewedig.com/2014/02/03/software-developers-reading-list/
This is a very practical book about Java multi-threading, and at the same time, it provides a lot of theoretical knowledge about concurrency in general. I highly recommend you read it at least once.
http://www.yegor256.com/2015/04/22/favorite-software-books.html

http://www.doolwind.com/blog/top-10-software-development-books/

If you understand most of them, your Java/Ruby/Python/Scala coding skills will improve significantly.
http://www.yegor256.com/2015/04/22/favorite-software-books.html
https://medium.com/javascript-scene/the-software-developer-s-library-a-treasure-trove-of-books-for-people-who-love-code-f9bc92c7883b#.4qktkub59

https://medium.com/javascript-scene/the-software-developer-s-library-a-treasure-trove-of-books-for-people-who-love-code-f9bc92c7883b#.4qktkub59

https://medium.com/javascript-scene/the-software-developer-s-library-a-treasure-trove-of-books-for-people-who-love-code-f9bc92c7883b#.4qktkub59

https://medium.com/javascript-scene/the-software-developer-s-library-a-treasure-trove-of-books-for-people-who-love-code-f9bc92c7883b#.4qktkub59

individual craftsmanship -- all the things that add up to what we instinctively call programming, maybe in college, but don't quite feel secure deciding what to do. software teams.
should know how to execute in any code base. Learning how to refactor your code can neatly divide my programming career into pre-Code Complete, pre-Clean Code and understand algorithms, which is the core of writing code.
are often seen in enterprise applications. This is a book I referenced all the time when describe it would be as an Anti-Dilbert Manifesto.
messier and harder to work with. Working with legacy code isn't fun, but this book software development. If you are a professional software developer, you must read doing and how a CPU actually executes your code. This is both a fun and fascinating on domain modeling. Once you learn what is in this book, you can’t go back. You won pattern. After reading this book it will be easy to memorize, reproduce and implement of a foundation you want to have. If you want to have a really solid foundation, then exercises.
tick and how they think. Definitely a must read!
personal choice for the most important first book.
Searching") stand out as true bibles of the industry.
http://www.yegor256.com/2015/04/22/favorite-software-books.html
build high quality software in short time. It has listed many best practices that can be principles contained in the book will lead to cleaner, more beautiful code.
for anyone who designs and develops computer software.
way that could be understood by both technical and non-technical audiences alike.
I almost never work with pure Java anymore, instead largely using other JVM-compatible languages, but the Java I wrote before reading Effective Java looks very different than the Java I wrote afterwards, and I definitely prefer the latter.
http://www.nomachetejuggling.com/2014/02/05/top-10-career-changing-programming-books/

http://www.infoworld.com/article/2620356/development-tools/10-must-read-books-for-developers.html

You may not always agree with the author but it provides good food for thought. It might be not what you expect, but may just be what you need.
https://dzone.com/articles/must-read-book-list-for-programmers

https://stevewedig.com/2014/02/03/software-developers-reading-list/

It's worth reading and re-reading if you're working with systems integration projects or writing integration software yourself.
http://www.fromdev.com/2010/08/best-software-architecture-books-must.html

https://stevewedig.com/2014/02/03/software-developers-reading-list/

https://medium.com/javascript-scene/the-software-developer-s-library-a-treasure-trove-of-books-for-people-who-love-code-f9bc92c7883b#.4qktkub59

this book, the way I wrote my code and the way I thought about writing code completely
otherwise would take years or even decades to learn.
aren't. They are inside this book. If you want a very practical programming book about
software development. The book explains numerous refactoring methods to eliminate smell
been explained in other languages. It is a great book for object orientated coding.
for college students learning programming courses.
apps, but that are applicable to more than one platform.
https://www.infoq.com/articles/recommended_reading_list
inherited codebase. If I'd read this book earlier, my first job experience would have been
suitable for people with a lot of OOP under their belts.
Petzold deals with a number of programming concepts starting from number systems -
objects at work.
know that not everyone is crazy about the Head First series, and even I find the structure and
provides simple solutions to complex programming. The book further explains the four trends
through the problems on your own (without looking ahead) you'll learn a lot and be a much
programming or best practices; this book gives developers a great opportunity to connect at a
https://www.reddit.com/r/learnprogramming/wiki/books
coverage of the topic.
https://stevewedig.com/2014/02/03/software-developers-reading-list/
http://www.doolwind.com/blog/top-10-software-development-books/
http://www.doolwind.com/blog/top-10-software-development-books/
http://www.philosophicalgeek.com/2007/11/21/books/
colors, pageflows and professional web design, all in an easy to read cover. Opening this

This book is packed with tips on how to write better code, be it concurrency, serialization or
other patterns to make your Java programs shine.
http://www.codepancake.com/10-books-every-programmer-should-read/

https://medium.com/javascript-scene/the-software-developer-s-library-a-treasure-trove-of-books-for-people-who-love-code-f9bc92c7883b#.4qktkub59

https://medium.com/javascript-scene/the-software-developer-s-library-a-treasure-trove-of-books-for-people-who-love-code-f9bc92c7883b#.4qktkub59

variables and statements. It helps to improve software craftsmanship
Moreover, it provides many concepts to develop high quality code,
various developments in software tools and environment for the last
designed code and turning it into something even humans can
working through lots of real-world examples.
structures. Comprehensive and quintessentially useful.
problems, and digs into solution details for each pattern.
put together a project that actually gets work done.
"Write unit tests, refactor code, make sure tests are passing."
patterns.
industry, developer or not.
for clearly expressing the abilities and limitations of the software, and
software. The trade-off for each patterns have been clearly pointed
ensuring the pieces come back to build the whole.
http://www.codepancake.com/10-books-every-programmer-should-
interviews. A really good starting point if you are curious about life as
library-a-treasure-trove-of-books-for-people-who-love-code-
library-a-treasure-trove-of-books-for-people-who-love-code-
library-a-treasure-trove-of-books-for-people-who-love-code-

http://feltpresence.com/reading_list

http://www.infoworld.com/article/2620356/development-tools/10-must-read-books-for-
http://www.infoworld.com/article/2620356/development-tools/10-must-read-books-for-
every developer.
even think you have to actually read it: your IDE probably supports many of the operations it
important! In Clean Code, “Uncle Bob” Martin shares tips and examples on how to create
http://www.codingdojo.com/blog/9-best-programming-books-read-right-now-want-distinguish/
you'd want to use (or avoid) a particular pattern. I can't tell you how many times I've
authors explain how managers can enable their software teams to realize their potential in a
https://blog.newrelic.com/2015/03/19/best-books-software-developer/
your own problems.
the underlying hardware. This book is designed to demystify the connection.
https://stevewedig.com/2014/02/03/software-developers-reading-list/
engaging style. The fact that it is a fun book to read is a major plus!
books-for-people-who-love-code-f9bc92c7883b#.4qktkub59
books-for-people-who-love-code-f9bc92c7883b#.4qktkub59
books-for-people-who-love-code-f9bc92c7883b#.4qktkub59

change the way you think about and write code.
dimensions that will help strengthen your programming career and
understand things that can go wrong in software development and will
https://dzone.com/articles/must-read-book-list-for-programmers
(fellow programmers) through code itself
precisely why an algorithm works in a certain manner.
the design of enterprise applications. For anyone venturing into the
applications that use the skills of the people who create them to
pitfalls, and typical failures. Most of the code we're working on now is
http://www.doolwind.com/blog/top-10-software-development-books/
library-a-treasure-trove-of-books-for-people-who-love-code-
http://feltpresence.com/reading_list

book.
launched an entire publishing company. It's a big deal, if you've
development year after year. A must read for every project manager
term perspective about what constitutes good design. It is an easy
how to write good code, but also an effective way to develop and
consult or recall how certain algorithm worked/performed I like to grab
sets out to do: provide the reader with a catalog of proven enterprise
http://www.doolwind.com/blog/top-10-software-development-books/
https://stevewedig.com/2014/02/03/software-developers-reading-list/
library-a-treasure-trove-of-books-for-people-who-love-code-

their development abilities specifically related to writing code.
your personality as a programmer. It is filled with practical advice on
lists and is still used in college courses says a lot.
of software using object oriented languages. Fowler and his fellow
author personal experience and opinions.
book is called ‘Introduction’, don’t underestimate it’s level because it
useful collection of patterns. It reads as practically a cookbook for a

http://www.yegor256.com/2015/04/22/favorite-software-books.html

another a simple coder who does acceptable but uninspired work? This book helps

surprised to see that software management problems have not changed that much

http://khanlou.com/2016/06/resources-for-new-programmers/

will guide you step by step into a becoming a master programmer!

https://www.reddit.com/r/learnprogramming/wiki/books

http://www.doolwind.com/blog/top-10-software-development-books/

a lot from it. I haven't stopped learning from this book, and every time
influence you in someway or another, and although there's a lot of
Month is definitely worthy of that time.

https://stevewedig.com/2014/02/03/software-developers-reading-list/

https://stevewedig.com/2014/02/03/software-developers-reading-list/
library-a-treasure-trove-of-books-for-people-who-love-code-

developer, I read Steve McConnell's Code Complete all the way
growing movement of people, including myself, who are promoting
http://www.philosophicalgeek.com/2007/11/21/books/
http://www.doolwind.com/blog/top-10-software-development-books/
library-a-treasure-trove-of-books-for-people-who-love-code-

weekends with it’s 900 pages, every page it contains is packed with deal of practical advise. This book will make you think instead of just

http://www.doolwind.com/blog/top-10-software-development-books/

library-a-treasure-trove-of-books-for-people-who-love-code-

http://www.philosophicalgeek.com/2007/11/21/books/

http://www.doolwind.com/blog/top-10-software-development-books/

library-a-treasure-trove-of-books-for-people-who-love-code-

http://www.doolwind.com/blog/top-10-software-development-books/
library-a-treasure-trove-of-books-for-people-who-love-code-

[library-a-treasure-trove-of-books-for-people-who-love-code-]