Programming Reading List

Credits to /u/CSMastermind for creating the list found here:

- Part 1
- Part 2
- Part 3

More reading list resources:

- The Ultimate Reading List for Developers

Job Interview Prep

- Cracking the Coding Interview: 189 Programming Questions and Solutions
- Programming Interviews Exposed: Coding Your Way Through the Interview
- Introduction to Algorithms
- The Algorithm Design Manual
- Effective Java
- Concurrent Programming in Java: Design Principles and Pattern
- Modern Operating Systems
- Programming Pearls
- Discrete Mathematics for Computer Scientists

Junior Software Engineer Reading List

Read This First

- Pragmatic Thinking and Learning: Refactor Your Wetware

Fundamentals

- Code Complete: A Practical Handbook of Software Construction
- Software Estimation: Demystifying the Black Art
- Software Engineering: A Practitioner's Approach
- Refactoring: Improving the Design of Existing Code
- Coder to Developer: Tools and Strategies for Delivering Your Software
- Perfect Software: And Other Illusions about Testing
- Getting Real: The Smarter, Faster, Easier Way to Build a Successful Web Application

Understanding Professional Software Environments

- Agile Software Development: The Cooperative Game
- Software Project Survival Guide
- The Best Software Writing I: Selected and Introduced by Joel Spolsky
- Debugging the Development Process: Practical Strategies for Staying Focused, Hitting Ship Dates, and Building Solid Teams
- Rapid Development: Taming Wild Software Schedules
- Peopleware: Productive Projects and Teams

Mentality

- Slack: Getting Past Burnout, Busywork, and the Myth of Total Efficiency
- Against Method
- The Passionate Programmer: Creating a Remarkable Career in Software Development

History

- The Mythical Man-Month: Essays on Software Engineering
- Computing Calamities: Lessons Learned from Products, Projects, and Companies That Failed
- The Deadline: A Novel About Project Management

Mid Level Software Engineer Reading List

Read This First

- Personal Development for Smart People: The Conscious Pursuit of Personal Growth

Fundamentals

- The Clean Coder: A Code of Conduct for Professional Programmers
- Clean Code: A Handbook of Agile Software Craftsmanship
- Solid Code
- Code Craft: The Practice of Writing Excellent Code
- Software Craftsmanship: The New Imperative
- Writing Solid Code

Software Design

- Head First Design Patterns: A Brain-Friendly Guide
- Design Patterns: Elements of Reusable Object-Oriented Software
- Domain-Driven Design: Tackling Complexity in the Heart of Software
- Domain-Driven Design Distilled
- Design Patterns Explained: A New Perspective on Object-Oriented Design
- Design Patterns in C# - Even though this is specific to C# the pattern can be used in any OO language.
- Refactoring to Patterns

Software Engineering Skill Sets

- Building Microservices: Designing Fine-Grained Systems
- Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools
- NoEstimates: How To Measure Project Progress Without Estimating
- Object-Oriented Software Construction
- The Art of Software Testing
- Release It!: Design and Deploy Production-Ready Software
- Working Effectively with Legacy Code
- Test Driven Development: By Example

Databases

- Database System Concepts
- Database Management Systems
- Foundation for Object / Relational Databases: The Third Manifesto
- Refactoring Databases: Evolutionary Database Design
- Data Access Patterns: Database Interactions in Object-Oriented Applications

User Experience

- Don't Make Me Think: A Common Sense Approach to Web Usability
- The Design of Everyday Things
- Programming Collective Intelligence: Building Smart Web 2.0 Applications
- User Interface Design for Programmers
- GUI Bloopers 2.0: Common User Interface Design Don'ts and Dos

Mentality

- The Productive Programmer
- Extreme Programming Explained: Embrace Change
- Coders at Work: Reflections on the Craft of Programming
- Facts and Fallacies of Software Engineering

History

- Dreaming in Code: Two Dozen Programmers, Three Years, 4,732 Bugs, and One Quest for Transcendent Software
- New Turning Omnibus: 66 Excursions in Computer Science
- Hacker's Delight
- The Alchemist
- Masterminds of Programming: Conversations with the Creators of Major Programming Languages

- The Information: A History, A Theory, A Flood

Specialist Skills

In spite of the fact that many of these won't apply to your specific job I still recommend reading them for the insight, they'll give you into programming language and technology design.

- Peter Norton's Assembly Language Book for the IBM PC
- Expert C Programming: Deep C Secrets
- Enough Rope to Shoot Yourself in the Foot: Rules for C and C++ Programming
- The C++ Programming Language
- Effective C++: 55 Specific Ways to Improve Your Programs and Designs
- More Effective C++: 35 New Ways to Improve Your Programs and Designs
- More Effective C#: 50 Specific Ways to Improve Your C#
- CLR via C#
- Mr. Bunny's Big Cup o' Java
- Thinking in Java
- JUnit in Action
- Functional Programming in Scala
- The Art of Prolog: Advanced Programming Techniques
- The Craft of Prolog
- Programming Perl: Unmatched Power for Text Processing and Scripting
- Dive into Python 3
- why's (poignant) guide to Ruby

Senior Level Software Engineer Reading List

Read This First

- Mastery: The Keys to Success and Long-Term Fulfillment

Fundamentals

- Patterns of Enterprise Application Architecture
- Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions
- Enterprise Patterns and MDA: Building Better Software with Archetype Patterns and UML
- Systemantics: How Systems Work and Especially How They Fail
- Rework
- Writing Secure Code
- Framework Design Guidelines: Conventions, Idioms, and Patterns for Reusable .NET Libraries

Development Theory

- Growing Object-Oriented Software, Guided by Tests
- Object-Oriented Analysis and Design with Applications
- Introduction to Functional Programming
- Design Concepts in Programming Languages
- Code Reading: The Open Source Perspective
- Modern Operating Systems
- Extreme Programming Explained: Embrace Change
- The Elements of Computing Systems: Building a Modern Computer from First Principles
- Code: The Hidden Language of Computer Hardware and Software

Philosophy of Programming

- Making Software: What Really Works, and Why We Believe It
- Beautiful Code: Leading Programmers Explain How They Think
- The Elements of Programming Style
- A Discipline of Programming
- The Practice of Programming
- Computer Systems: A Programmer's Perspective
- Object Thinking
- How to Solve It by Computer

- 97 Things Every Programmer Should Know: Collective Wisdom from the Experts

Mentality

- Hackers and Painters: Big Ideas from the Computer Age
- The Intentional Stance
- Things That Make Us Smart: Defending Human Attributes In The Age Of The Machine
- The Back of the Napkin: Solving Problems and Selling Ideas with Pictures
- The Timeless Way of Building
- The Soul Of A New Machine
- WIZARDRY COMPILED
- YOUTH
- Understanding Comics: The Invisible Art

Software Engineering Skill Sets

- Software Tools
- UML Distilled: A Brief Guide to the Standard Object Modeling Language
- Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development
- Practical Parallel Programming
- Past, Present, Parallel: A Survey of Available Parallel Computer Systems
- Mastering Regular Expressions
- Compilers: Principles, Techniques, and Tools
- Computer Graphics: Principles and Practice in C
- Michael Abrash's Graphics Programming Black Book
- The Art of Deception: Controlling the Human Element of Security
- SOA in Practice: The Art of Distributed System Design
- Data Mining: Practical Machine Learning Tools and Techniques
- Data Crunching: Solve Everyday Problems Using Java, Python, and more.

Design

- The Psychology Of Everyday Things
- About Face 3: The Essentials of Interaction Design
- Design for Hackers: Reverse Engineering Beauty
- The Non-Designer's Design Book

History

- Micro-ISV: From Vision to Reality
- Death March
- Showstopper! the Breakneck Race to Create Windows NT and the Next Generation at Microsoft
- The PayPal Wars: Battles with eBay, the Media, the Mafia, and the Rest of Planet Earth The Business of Software: What Every Manager, Programmer, and Entrepreneur Must Know to Thrive and Survive in Good Times * and Bad
- In the Beginning...was the Command Line

Specialist Skills

- The Art of UNIX Programming
- Advanced Programming in the UNIX Environment
- Programming Windows
- Cocoa Programming for Mac OS X
- Starting Forth: An Introduction to the Forth Language and Operating System for Beginners and Professionals
- lex & yacc
- The TCP/IP Guide: A Comprehensive, Illustrated Internet Protocols Reference
- C Programming Language
- No Bugs!: Delivering Error Free Code in C and C++
- Modern C++ Design: Generic Programming and Design Patterns Applied
- Agile Principles, Patterns, and Practices in C#
- Pragmatic Unit Testing in C# with NUnit

DevOps Reading List

- Time Management for System Administrators: Stop Working Late and Start Working Smart
- The Practice of Cloud System Administration: DevOps and SRE Practices for Web Services
- The Practice of System and Network Administration: DevOps and other Best Practices for Enterprise IT
- Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale
- DevOps: A Software Architect's Perspective
- The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations
- Site Reliability Engineering: How Google Runs Production Systems
- Cloud Native Java: Designing Resilient Systems with Spring Boot, Spring Cloud, and Cloud Foundry
- Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation
- Migrating Large-Scale Services to the Cloud

Entrepreneur Reading List

- Disrupted: My Misadventure in the Start-Up Bubble
- The Phoenix Project: A Novel about IT, DevOps, and Helping Your Business Win
- The E-Myth Revisited: Why Most Small Businesses Don't Work and What to Do About It
- The Art of the Start: The Time-Tested, Battle-Hardened Guide for Anyone Starting Anything
- The Four Steps to the Epiphany: Successful Strategies for Products that Win
- Permission Marketing: Turning Strangers into Friends and Friends into Customers
- Ikigai
- Reality Check: The Irreverent Guide to Outsmarting, Outmanaging, and Outmarketing Your Competition
- Bootstrap: Lessons Learned Building a Successful Company from Scratch
- The Marketing Gurus: Lessons from the Best Marketing Books of All Time
- Content Rich: Writing Your Way to Wealth on the Web
- The Web Startup Success Guide
- The Best of Guerrilla Marketing: Guerrilla Marketing Remix
- From Program to Product: Turning Your Code into a Saleable Product This Little Program Went to Market: Create, Deploy, Distribute, Market, and Sell Software and More on the Internet at * Little or No Cost to You
- The Secrets of Consulting: A Guide to Giving and Getting Advice Successfully
- The Innovator's Solution: Creating and Sustaining Successful Growth
- Startups Open Sourced: Stories to Inspire and Educate
- In Search of Stupidity: Over Twenty Years of High Tech Marketing Disasters
- Do More Faster: TechStars Lessons to Accelerate Your Startup Content Rules: How to Create Killer Blogs, Podcasts, Videos, Ebooks, Webinars (and More) That Engage Customers and Ignite * Your Business
- Maximum Achievement: Strategies and Skills That Will Unlock Your Hidden Powers to Succeed
- Founders at Work: Stories of Startups' Early Days
- Blue Ocean Strategy: How to Create Uncontested Market Space and Make Competition Irrelevant
- Eric Sink on the Business of Software
- Words that Sell: More than 6000 Entries to Help You Promote Your Products, Services, and Ideas
- Anything You Want
- Crossing the Chasm: Marketing and Selling High-Tech Products to Mainstream Customers
- The Innovator's Dilemma: The Revolutionary Book that Will Change the Way You Do Business
- Tao Te Ching
- Philip & Alex's Guide to Web Publishing
- The Tao of Programming
- Zen and the Art of Motorcycle Maintenance: An Inquiry into Values
- The Inmates Are Running the Asylum: Why High Tech Products Drive Us Crazy and How to Restore the Sanity

Computer Science Grad School Reading List

- All the Mathematics You Missed: But Need to Know for Graduate School
- Introductory Linear Algebra: An Applied First Course
- Introduction to Probability
- The Structure of Scientific Revolutions
- Science in Action: How to Follow Scientists and Engineers Through Society
- Proofs and Refutations: The Logic of Mathematical Discovery
- What Is This Thing Called Science?
- The Art of Computer Programming
- The Little Schemer
- The Seasoned Schemer

- Data Structures Using C and C++
- Algorithms + Data Structures = Programs
- Structure and Interpretation of Computer Programs
- Concepts, Techniques, and Models of Computer Programming
- How to Design Programs: An Introduction to Programming and Computing
- A Science of Operations: Machines, Logic and the Invention of Programming
- Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology
- The Computational Beauty of Nature: Computer Explorations of Fractals, Chaos, Complex Systems, and Adaptation
- The Annotated Turing: A Guided Tour Through Alan Turing's Historic Paper on Computability and the Turing Machine
- Computability: An Introduction to Recursive Function Theory
- How To Solve It: A New Aspect of Mathematical Method
- Types and Programming Languages
- Computer Algebra and Symbolic Computation: Elementary Algorithms
- Computer Algebra and Symbolic Computation: Mathematical Methods
- Commonsense Reasoning
- Using Language
- Computer Vision
- Alice's Adventures in Wonderland
- Gödel, Escher, Bach: An Eternal Golden Braid

Video Game Development Reading List

- Game Programming Gems - 1 2 3 4 5 6 7
- AI Game Programming Wisdom - 1 2 3 4
- Making Games with Python and Pygame
- Invent Your Own Computer Games With Python
- Bit by Bit