

[Sign Up For Future Free Content, Videos, Etc](#)

Intro

This is a software engineering study guide that you can use to help prepare yourself for your interview. This was developed by people who have interviewed and gotten jobs at Amazon, Facebook, Capital One and several other tech companies. We hope these help you get great jobs as well.

In order to use this, you can make a copy of this sheet and follow along with the study guide. Keeping track helps you know where you are and how you are doing.

Warm Up With The Classics

[Fizz Buzz](#)

[Find The Kth Smallest/Largest Integer In An Array](#)

[560. Subarray Sum Equals K](#)

[Arrays: Left Rotation](#)

[Strings: Making Anagrams](#)

[Nth Fibonacci](#)

How did you? Take a moment and rate yourself on these classics. We have been asked most of these at some point in the interview process and often early on as a weed-out style questions. They often have less to do with algorithms and data structures, but still require a good understanding of loops and arrays (yes an array is a data structure).

Algorithms And Data Structures

Pre-Study Problems

Before going through the video content about data structures and algorithms. Consider trying out these problems below. See if you can answer them. This will help you know what to focus on.

[985. Sum of Even Numbers After Queries](#)

[657. Robot Return to Origin](#)

[961. N-Repeated Element in Size 2N Array](#)

[110. Balanced Binary Tree](#)

[3. Longest Substring Without Repeating Characters](#)

[19. Remove Nth Node From End of List](#)

[23. Merge k Sorted Lists](#)

[31. Next Permutation](#)

Algorithms And Data Structures Videos

Data Structures

[Data Structures & Algorithms #1 - What Are Data Structures?](#)

[Multi-dim \(video\)](#)

[Dynamic Arrays \(video\)](#)

[Resizing arrays \(video\)](#)

[Data Structures: Linked Lists](#)

[Core Linked Lists Vs Arrays \(video\)](#)

Pointers to Pointers
Data Structures: Trees
Data Structures: Heaps
Data Structures: Hash Tables
Phone Book Problem (video)
Data Structures: Stacks and Queues
Using Stacks Last-In First-Out (video)
Data Structures: Crash Course Computer Science #14
Data Structures: Tries
Algorithms
Python Algorithms for Interviews
Algorithms: Graph Search, DFS and BFS
BFS(breadth-first search) and DFS(depth-first search) (video)
Algorithms: Binary Search
Binary Search Tree Review (video)
Algorithms: Recursion
Algorithms: Bubble Sort
Algorithms: Merge Sort
Algorithms: Quicksort
Big O Notation
Introduction to Big O Notation and Time Complexity (Data Structures & Algorithms #7)
Harvard CS50 - Asymptotic Notation (video)
A Gentle Introduction to Algorithm Complexity Analysis
Cheat sheet
Dynamic Programming
Dynamic Programming (Think Like a Programmer) - Video
Algorithms: Memoization and Dynamic Programming - Video
6.006: Dynamic Programming I: Fibonacci, Shortest Paths - Video
6.006: Dynamic Programming II: Text Justification, Blackjack - Video
6.046: Dynamic Programming & Advanced DP - Video
Dynamic Programming - Post
String Manipulation
Coding Interview Question and Answer: Longest Consecutive Characters
Sedgewick - Substring Search (videos)
Interview Problem Walk Throughs
Amazon Coding Interview Question - Recursive Staircase Problem
Google Coding Interview - Universal Value Tree Problem
Google Coding Interview Question and Answer #1: First Recurring Character
Find min and max element in a binary search tree (video)

[Find height of a binary tree \(video\)](#)

[Check if a binary tree is binary search tree or not \(video\)](#)

[What Is Tail Recursion Why Is It So Bad?](#)

Post-Study Problems

Now that you have studied for a bit, and watched a few videos. Let's try some more problems!

[Bigger Is Greater](#)

[6. ZigZag Conversion](#)

[7. Reverse Integer](#)

[40. Combination Sum II](#)

[43. Multiply Strings](#)

[Larry's Array](#)

[Short Palindrome](#)

[65. Valid Number](#)

[Bigger is Greater](#)

[The Full Counting Sort](#)

[Lily's Homework](#)

[Common Child](#)

[459. Repeated Substring Pattern](#)

[27. Remove Element](#)

[450. Delete Node in a BST](#)

[659. Split Array into Consecutive Subsequences](#)

[Number of Subarrays with Bounded Maximum](#)

[Combination Sum IV](#)

[Best Time to Buy and Sell Stock with Cooldown](#)

[Longest Repeating Character Replacement](#)

[Swap Nodes in Pairs](#)

[Binary Tree Right Side View](#)

[Flatten Nested List Iterator](#)

[Binary Tree Level Order Traversal](#)

[Binary Search Tree Iterator](#)

[Maximum Length of Pair Chain](#)

[Split Linked List in Parts](#)

Operational Programming Problems

Some companies won't ask you algorithm problems. Instead, they might focus more on implementation and operational problems. These are usually more niche and involve practical problems. Like looping through data and performing a task of some sort. These types of problems don't usually require as much practice because it is more about taking basic concepts like arrays and hashmaps and keeping track of what you are doing to them.

[Kangaroo Problem](#)

[Breaking Records](#)

[Find A String](#)[itertools.permutations\(\)](#)[No Idea!](#)[Days of the programmer](#)[Leaderboard](#)[Word Order](#)[Sherlock And Squares](#)[Equalize The Array](#)[Apples And Oranges](#)[More Operational Style Questions](#)

System Design Videos

System design are crucial questions that show you are more than just a coder. You need to be able to think big picture as an engineer. Where do certain services belong, what kind of servers do you need, how would you manage traffic, etc. All of these ideas show that you are able to design software not just code what people tell you to code.

[Tiny url](#)[Parking Lot System](#)[Whats App](#)[Uber design](#)[Instagram](#)[Tinder Service](#)

Operating Systems

Operating system questions are a little more rare, but it is good to have a good basis in understand concepts like threads, scheduling, memory, etc. Even if it is just a basic understanding. It is very embarrassing to get asked what is the difference between a process and a thread and not know the answer.

[Commonly Asked Operating Systems Interview Questions](#)[What is Translation lookaside buffer?](#)[Why does Round Robin avoid the Priority Inversion Problem?](#)[Interrupt Vs System Call](#)[What is 'inode' in file system?](#)[Operating System Interview Questions and Answers - Part I](#)[What is a kernel - Gary explains](#)[Round Robin Algorithm Tutorial \(CPU Scheduling\)](#)[The Magic of LRU Cache \(100 Days of Google Dev\) \(video\)](#)[MIT 6.004 L15: The Memory Hierarchy \(video\)](#)[Interrupts \(video\)](#)[Scheduling \(video\)](#)

Threads

[User Level thread Vs Kernel Level thread](#)

[Intro to Processes & Threads](#)

[Difference Between Process and Thread - Georgia Tech - Advanced Operating Systems](#)

[Difference between forking and multithreading](#)

Object Oriented

Similar to operating systems, not every interview will ask you about object oriented programming, but you never know. You want to make sure you remember your basics from your computer 162 course.

[Java Programming Tutorial - 49 - Inheritance](#)

[Java Programming Tutorial - 55 - Introduction to Polymorphism](#)

[Java Programming Tutorial - 58 - Abstract and Concrete Classes](#)

[Java Programming Tutorial - 57 - Overriding Rules](#)

[Java Programming Tutorial - 59 - Class to Hold Objects](#)

[Object-Oriented Programming](#)

Design Patterns

If you were like us, we weren't taught about all the various design patterns. So it's good to get an understanding of how they work and why you would use them. Some interview questions can be as simple as why would you use a factory class.

[Factory Design Pattern](#)

[Observer Design Pattern](#)

[Adapter Design Pattern](#)

[Facade Design Pattern](#)

[Chain of Responsibility Design Pattern](#)

[Interpreter Design Pattern](#)

[Singleton Design Pattern Tutorial](#)

[Chapter 6 \(Part 1\) - Patterns \(video\)](#)

[Head First Design Patterns](#)

SQL - Problems

[262. Trips and Users](#)

[601. Human Traffic of Stadium](#)

[185. Department Top Three Salaries](#)

[626. Exchange Seats](#)

[Hackerrank The Report](#)

[177. Nth Highest Salary](#)

[Symmetric Pairs](#)

[Occupations](#)

[Placements](#)

[Ollivander's Inventory](#)

SQL - Videos

[IQ15: 6 SQL Query Interview Questions](#)

[Learning about ROW_NUMBER and Analytic Functions](#)

Advanced Implementation Of Analytic Functions
Advanced Implementation Of Analytic Functions Part 2
Wise Owl SQL Videos
Post SQL Problems
Binary Tree Nodes
Weather Observation Station 18
Challenges
Print Prime Numbers
595. Big Countries
626. Exchange Seats
SQL Interview Questions: 3 Tech Screening Exercises (For Data Analysts)
Courses
If you really don't feel comfortable, then consider taking one of the courses below. It will pretty much have the same videos and discussions above. However, sometimes paying for courses is a good incentive to actually get it done.
The Coding Interview Bootcamp: Algorithms + Data Structures
Data Structures and Algorithms Bootcamp
Data Structures and Algorithms - The Complete Masterclass
Books
Some of us hate reading books and watching online videos. For you few, we have some classic book recommendations for preparing for software engineering interview.
Cracking The Code
The Algorithm Design Manual
The Linux Command Line: A Complete Introduction
The Complete Software Developer's Career Guide: How to Learn Your Next Programming Language, Ace Your Programming Interview, and Land The Coding Job Of Your Dreams

Other Resources

[Learning Data Science: Our Top 25 Data Science Courses](#)
[The Best And Only Python Tutorial You Will Ever Need To Watch](#)
[Dynamically Bulk Inserting CSV Data Into A SQL Server](#)
[4 Must Have Skills For Data Scientists](#)
[Engineering Dashboards, Metrics And Algorithms Part 2](#)
[Read Last Weeks Top Ten Article For Python Libraries](#)
[How Algorithms Can Become Unethical and Biased](#)
[SQL Best Practices — Designing An ETL Video](#)

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

Personal Difficulty 1-5[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]