How to ace a technical interview at Amazon.

Congratulations on securing an interview with Amazon Video!

This document aims to help prepare candidates for a Software development Engineer (SDE) interview.

In your interview you will have four, one hour interviews. In each interview you will be asked one technical question (with extensions) and two leadership, or behavioural, interview questions.

You will usually spend about 40 minutes on the technical question and 15 minutes on the leadership questions leaving a little time for questions at the end.

Technical interview questions at Amazon can be broken down into three “types”

Algorithms & Data structures
Coding / OO design
Systems Design

A few preliminary notes:

- Interviewers may offer tips/hints, listen closely for these.
- We will never ask a trick question or deliberately mislead you in anyway.
- Be vocal throughout.
- You Will **NOT** be asked questions about being stuck in a blender or about how many windows there are in New York (City or otherwise).

LEADERSHIP QUESTIONS

Our Leadership Principles aren't just a pretty inspirational wall hanging. These Principles work hard, just like we do. Amazonians use them, every day, whether they're discussing ideas for new projects, deciding on the best solution for a customer's problem, or interviewing candidates. It's just one of the things that makes Amazon peculiar.

Leadership questions are based on Amazons leadership principles and will usually be structured: "Tell me about a time when…" OR "Give me an example of…"

A full list of our principles can be found here. Get to know these and prepare some examples of where you have shown these attributes.

Your examples can be from commercial experience, your own ventures, education, hackathons, open source projects or any other area relevant to software engineering or computer science.

Preparation for this part of the interview is somewhat limited as it's based on your current experience, you can however prepare by gathering and memorizing data to back up your examples and create a good structure to your examples. We love data here at Amazon whether it be good or bad we can usually learn something from it.

amazon prime video

ALGORITHMS & DATA STRUCTURES

Put simply … design an algorithm to solve a particular problem.

It usually starts like this:

“Given X, figure out an efficient way to do Y.”

Make sure you understand the problem. You’re not going to lose points asking for clarifications or talking through the obvious upfront. We want to see you being meticulous in diving into the problem and understanding it. In your day to day work you would not try to solve a problem without understanding it fully first, we don’t want you to do this in interview either. Ask as many questions as you like, the interviewer is there to help you as much as evaluate your skills.

Once you understand the problem, **keep an eye on the time** and don't give in to paralysis by analysis! Try to come up with a solution. If needs be, start with something trivial, or inefficient. If you can achieve a brute force solution to a problem, you’ve cleared a major hurdle to solving it in a more efficient way.

Talk it through, we want to hear your thought process no matter how bizarre! Vocalising is a great thing especially under timed conditions where there is always a chance of running out of time. It can mean the difference between “we ran out of time and I’m not sure where John was going with his skip list” and “although we ran out of time, what we spoke about was beautiful and given another 10 minutes I know we would have reached an optimal solution.”

Break it down into smaller problems. Look at specific use case, edge and corner cases. Or a simplified version of the problem and work your way up to the full scope.

Prepare to be asked about more commonly used Algorithms & Data Structures.

Don't be afraid to change direction. If while working towards a solution you suddenly think of a better way or you feel like a particular approach isn't working, speak about it with the interviewer. Try a different approach. Most the time our problems have many different solutions, much like our day to day work.

THE CODING INTERVIEW/ OO DESIGN

We need problem solvers, but also engineers who know how to write solid, production-quality code.

In the comfort of your preferred IDE you may be thinking no problem….how about on a whiteboard, under interview conditions, on an unfamiliar problem? We realise a coding interview is far removed from real life and we adjust our expectations accordingly. We are not expecting a product we could ship tomorrow, instead we are looking for evidence of an ability to write correct and clean code that can be used by your team and other teams within Amazon.

As you work, we are not only looking at the syntax but are also watching for your ability to test your problem-solving/ analytical skills and creativity.

BEFORE YOU START

Again, make sure you **understand the problem**. Don't hesitate to ask questions. Specifically, follow up if any of the problem requirements seem loosely defined or otherwise unclear. There is no penalty for asking for clarifications, and you don't want to miss a key requirement or proceed on unfounded assumptions.

Pick a language. We don't have a preference for which language you use. As long as you have a firm grasp on the fundamentals (decomposition, object-oriented design, etc.) and are happy to be adaptable when needs be we believe you can pick up a new language fairly quickly.

Be vocal. Explain your thought process to your interviewer as you code. This helps you more fully communicate your solution, and gives your interviewer an opportunity to correct misconceptions or provide high-level guidance/feedback.

Be aware of and ready to talk about edge cases. Naturally, you should strive for a solution that's correct in all observable aspects. Sometimes there will be a flaw in the core logic of your solution, but more often your only bugs will be in how you handle edge cases (this is true of real-world engineering as well). Make sure your solution works on all edge cases you can think of.

Discuss any shortfalls in your solution, have you taken any shortcuts or done anything differently than you would in a real world example? Maybe there is another solution you decided not to proceed with. Why?

SYSTEMS DESIGN

If you are a graduate or have 1-2 years in industry we do not expect you to be a master in systems design, however you may be asked a light version of a systems design question to assess your problem solving and creativity.

A truly unbelievable amount of complexity lies beneath something as simple as streaming the latest episode of The Grand Tour or suggesting you may enjoy The Man in the High Castle" While most of that complexity is abstracted away from the end user, as an engineer at Amazon you would own this complexity.

At Amazon you would rarely be asked to implement fully-defined features. Instead we take ownership of *open-ended and ambiguous problems*, and it's our job to come up with the best solution to each. To do this you must be able to communicate effectively with the people around you. Working on problems with huge scope isn't something you will want to do alone.

Usually we'll start by asking you a wildly open ended question. These problems are bottomless. You will need to guide the conversation and understand the problem. That means asking questions, sketching diagrams on the board, and bouncing ideas off your interviewer. Do you know the constraints? What kind of inputs does your system need to handle? You have to get a sense for the scope of the problem before you start exploring the space of possible solutions. And remember, there is not always a single right answer.

Think about **Concurrency, Networking, Abstraction, Real-World Performance, Estimation, Availability and Reliability.**

The systems design interview can be challenging, but it's also a place to be creative and to take joy in the imagining of systems unbuilt. If you listen carefully, make sure you fully understand the problem, and then take a clear, straightforward approach to communicating your ideas you will already be half way there.

On a **final note** we really appreciate you taking the time out to speak with us and want you to enjoy the interview process as much as an interview can possibly be enjoyed.

If there is anything you are unsure about or anything we can do to make your visit more enjoyable just let your recruiter know.

Good luck!