Technical Interviews

What is a Technical Interview?
Technical Interviews are common when applying for positions in tech, IT, engineering, data analysis, and science. They feature questions that are specific to the role you have applied for, so that the employer can confirm you have the requisite skills. Depending on the job you're interviewing for, you will be asked about the skills, experience, certifications, competencies, languages, processes, systems, and tools you have that are a match for the job requirements.

Possible Interview Components

Live coding
- Usually happens on an IDE or other platform like Google Docs
- Can be in-person or virtual
- Interviewees are usually given between one to three questions to solve (depending on difficulty and time availability) to solve while communicating their thought processes

Coding on whiteboard/paper
- Done by hand and in-person
- As with live coding interviews, interviewees are usually given between one to three questions to solve (depending on difficulty and time availability) while communicating their thought processes

General questions
- Tests interviewees’ fundamental computer science knowledge without coding
- May include questions like how to implement a queue using an arraylist vs. a linklist, how to define inheritance as an OOP principle, etc.

Tips

Before the interview
- Practice communicating your thought process out loud to others.
- Make sure you know your best technical language well. Practice using the main data structures in that language, space and time complexities for different data structures and algorithms, and OOP principles.
- Use the resources below to prepare for your technical interview as well as Google and Glassdoor to find out what interview questions are likely to be asked.
- Not sure if your upcoming job interview has a technical component? Ask your recruiter what to expect!
During the interview

- Consider your interviewer a colleague who you are collaborating with and to whom you’re explaining your approach.
- Think out loud about the tradeoffs for different approaches. Why did you choose to try what you’re doing?
- Don’t be afraid to communicate if you’re stuck. It’s helpful for interviewers to see how you work through difficult problems. Let the interviewer know if you need a second to think.
- A bulky solution that works is better than attempting an optimized solution that doesn’t work.
- Don’t ignore hints from your interviewers!

How Can I Prepare?

Written Resources
Books like *Cracking the Coding Interview* offer programming questions and solutions to help you anticipate what interviewers may ask.

Online Resources
Websites like LeetCode offer pools of coding questions and structured fundamentals training. They are great resources to simulate the environment of live-coding interviews. Online platforms like Pramp allow you to practice interview questions online with peers as an easy way to practice in a low-stakes environment.

White Board Practice
In order to get comfortable in the environment of a technical interview, it’s important to become familiar with their set-up. Practicing in front of a friend on a whiteboard will help you get better at communicating your thought processes as you work through coding problems.