



Technical Interview Resources

Lyft University Programs

Below are resources to help you prepare for technical interviews. You also will find advice from Lyft engineers as well as information about the general Lyft interview process. We look forward to working with you in the future!

Reading Recommendations:

- [Cracking the Coding Interview](#)
 - The *Interview and Beyond* section has general interview strategies, technical questions, and offer negotiations (**highly recommended**) - Prioritize Chapters 1-4, 8, 9, and 11. These cover core topics
- [Code Complete: A Practical Handbook of Software Construction](#)
 - Widely considered as one of the best practical programming guides - The Second Edition is fully updated and revised with new code samples

Online Coding Practice Tools:

- [HackerRank](#) - leading end-to-end hub of technical practice problems (**highly recommended**)
 - Collaborative & Hands-on experience with general programming interview-level questions
- [Leetcode](#) - versatile platform for enhancing fundamental coding skills (**highly recommended**)
 - One of the largest pools of coding questions, along with structured fundamentals training
- [Firecode.io](#) - daily technical interview practice with emphasis on fundamentals
- [CareerCup](#) - peer forum for interview questions and a mock interview service
- [Python Tutor](#) - programming fundamentals and code visualization
- [Pramp](#) - coding interview practices and live feedback & exercises
- [Algorist](#) - A helpful wiki page that you can use as a self-study guide and resource
 - We highly recommend studying the [data structures](#), [sorting and searching](#), [combinatorial search and heuristic methods](#), and [dynamic programming](#) chapters

Tips & Experiences

- Lyft Engineer's account on interviews: [blog post](#)
 - Note: intern interviews will not cover design & architecture
- We use [CoderPad](#) for our general technical interviews. ([Codepen](#) for Frontend interviews)
- Always look through the job descriptions on lyft.com/careers for expectations & background
- We are language-agnostic; code in your strongest language for best results
- If you get stuck during interviews, your interviewer is a resource to help you out
- Do your research and practice with the given resources; don't be afraid to ask questions

Process

- Your process will always begin with an initial assessment that covers CS fundamentals.
- If you pass the initial assessment, we will move onto a technical phone interview; typically 60- minutes long, and will be conducted via tools like CoderPad or Codepen (for gen SWE or Frontend, respectively).
- The final stage involves a programming challenge that tests your CS fundamentals as well as your ability to solve issues that are similar to real-world coding problems.