Technical Interviewing

You may be asked to participate in a technical interview if the employer would like to evaluate your technical skills as a part of the job application process. Typically, a phone or first-round interview will take place as part of the initial screening, and there will not be many technical questions. Advanced technical questioning will instead take place during the second round of face-to-face interviews.

Candidates should prep the following before an interview:

• Review core algorithms and computer science skills.
• Know what technical skills are on your resume.
• Practice writing code with sample questions and focus on your strongest language.
  o Practice on a white board or paper.
  o Do not use pseudo code. Employers will want to see you use actual code.
  o Write in the language in which you feel the most comfortable. If you feel comfortable with a few languages, use the one your interviewer also knows. During the interview, you can ask what your interviewer likes.
• Practice! Schedule an appointment with Career Services specifically for technical interviewing. If time is short, practice with a friend.
• If an interviewer asks you if you are familiar with a sorting algorithm, do not say yes unless you can demonstrate it on the whiteboard or on paper before her or him.

How to solve any technical interview coding question:

1. Clarify the question
2. Determine inputs and outputs
3. Determine edge cases
4. Determine the brute force solution
5. Optimize your solution BUD
  o Bottle necks?
  o Unnecessary work?
  o Duplicate work?
6. Test your solution
7. Code your solution
8. Walkthrough and explain your solution

Typical technical interviews will cover:

• Object orientation: Do you know what an object is and how to use them?
• Data structures: This consists largely of string manipulations and array problems
• Algorithmic structure: These include linked lists, bit manipulation, sorting, and searching.
• Testing: How would you test a website? A car?
• Databases: Think SQL. What is a relational database?

Interviewers expect you to:

• Be familiar with a language so you can talk about it and solve a coding problem during the interview. You can use any language you want as long as you know how to code in it.
• Explain your thought process and how you are solving the problem, i.e. talk it out. Expect to have a conversation with the interviewer (especially if it is a phone interview!). Do not remain quiet when trying to solve the problem. You should talk through the problems and have a conversation with the interviewer about both the benefits and disadvantages of different solutions to the problem.
• Spend as long as you need before moving to the whiteboard, and be very sure of what you are about to write. Remember: Until you know your input and your output, you are not ready to write code.
• If you are stuck, start with a simple solution and then upgrade to a more complex solution. A great engineer is someone who knows a little about a lot, and a lot about a little: Be comfortable in the computer science industry but know that you are not expected to be an expert. Bring the conversation back to what you know as often as you can.
• Go beyond giving the interviewer what he or she asked. A top-tier candidate talks about subtleties in design, data structures, and algorithms. You should also be able to offer a variety of alternatives in different situations.

**How to handle a 30-45 minute technical interview:**
• Short introduction: Be ready to talk about the technical skills with which you are most comfortable. This is not a time to exaggerate, but be confident in sharing your skills.
• Assessment question: Answer these questions the best you can. Remember that it is more important to get the high-level correct answer than to write code perfectly, but perfect code is a plus.
• Test your code: Stand back and make sure that what you wrote is correct. Show the interviewer that the performance and accuracy of your code is important to you.

**What to do when you have seen a question before:**
If you have heard a question before, admit this to your interviewer. Your interviewer is asking you these questions to evaluate your problem-solving skills. If you already know the question, then you are not giving them the opportunity to evaluate you. Additionally, your interviewer may find it highly dishonest if you do not reveal that you know the question and, conversely, the interviewer will view you with high regard and take note that you disclosed this information.

**Additional tips:**
• Realize that the interviewer is not out to trick you!
• Be relaxed, and be calm as you work through the technical problems.
• Be polite and sociable. The interviewers want to know if they can work with you.

**Sample questions:**
• Can you describe your technical contributions to a project on your resume?
• How would you sort a two dimensional array of integers on the 0th element?
• Write a method to replace all the spaces in a string with “%20”.
• How would you test a car?

**Resources for interview prep:**

**Websites**
• [HackerRank](https://www.hackerrank.com) (entry level website)
• [LeetCode](https://leetcode.com) (some of these actual questions could be used in interviews)
• [GeeksforGeeks](http://www.geeksforgeeks.org) (well-developed explanations of complicated topics)
• [Pramp](https://www.pramp.com) (gives practice phone interviews with real people)

**Books**
• [Cracking the Coding Interview](https://www.crackingthecodinginterview.com) by Gayle Laakmann McDowell
• [Introduction to Algorithms](https://mitpress.mit.edu/books/introduction-to-algorithms) by CLRS by Charles E. Leiserson, Clifford Stein, Ronald Rivest, and Thomas H. Cormen

**Additional Resources**
• [Hangout on Air](https://www.google.com/about/jobs/lifeatgoogle/hangout-on-air-tech- interviewing.html): Candidate coaching session—tech interviewing.