LUDDY SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING
CAREER SERVICES

Technical Interviewing

If an employer needs to understand your technical skills, you may be asked to participate in a technical interview. Typically, a virtual interview without many technical questions will be the first round interview. Advanced technical questioning usually takes place during the second round of 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.
    - Practice on a whiteboard or paper.
    - Do not use pseudo code. Employers will want to see you use actual code.
    - Write in the language in which you feel the most comfortable. If you feel comfortable with a few languages, use one your interviewer also knows. During the interview, you can ask what your interviewer prefers.
- **Practice technical interviewing with the Luddy Peers!** Set up a Technical Interview appointment in Handshake. If time is short, practice with a friend or online.
- If an interviewer asks if you're familiar with a sorting algorithm, do not say yes unless you are prepared to demonstrate it on the whiteboard or on paper.

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 with BUD:
    - **B**ottlenecks?
    - **U**nnecessary work?
    - **D**uplicate work?
6. Test your solution
7. Code your solution
8. Walk through and explain your solution

Typical technical interviews will cover:

- **Object Orientation:** Do you know what an object is and how to use it?
- **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. Use any language you like as long as you know how to code in that language.
- Explain your thought process and how you are solving the problem. Expect to have a conversation with the interviewer (especially if it is a virtual 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. You should be comfortable with coding 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 they asked. **A top-tier candidate talks about subtleties** in design, data structures, and algorithms. Tryto offer a variety of alternatives in different situations.

HOW TO HANDLE A 30–40 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. Your interviewer may find it highly dishonest if you do not reveal that you know the question!

Conversely, if you admit you already know the question, the interviewer will view you with high regard.

ADDITIONAL TIPS:

- **Remember that the interviewer is not trying to trick you!**
- Be relaxed and calm as you work through the technical problems.
- Be **polite and responsive.** 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** (entry level website): hackerrank.com
- **LeetCode** (some of these actual questions could be used in interviews): leetcode.com
- **GeeksforGeeks** (well-developed explanations of complicated topics): geeksforgeeks.org
- **Pramp** (practice virtual interviews with real people): pramp.com

Books

- **Cracking the Coding Interview** by Gayle Laakmann McDowell
- **Introduction to Algorithms** by CLRS by Charles E. Leiserson, Clifford Stein, Ronald Rivest, and Thomas H. Cormen

Additional Resources

- **LifeAtGoogle:** Prepare for Your Google Interview: Troubleshooting & Scripting youtu.be/_8cH-QPVXsw
- **LifeAtGoogle:** Prepare for Your Google Interview: Coding youtu.be/6ZZX9ilgFooIndiana