



Welcome, we're happy that you're considering Microsoft for the next step in your career!

We're providing this guide to provide you with insights into our style of technical interviews, what to expect and how to prepare.

Your day will start with a welcome and introduction, and you will proceed to four interviews, focusing on coding, design, and algorithms, whilst we look into Microsoft competencies as well.

## **On-Site Interview Highlights**

Our interviews evaluate candidates holistically in two ways: competencies and technical capability that align with the list below. We hope that the pointers below would be of help in your preparation for the interview.

## **Preparing for a Competency Interview**

- Competencies can be shown both whilst you answer the technical question, or in answering a question based on your experience.
- In answering questions based on experience, describe recent, significant events that represent the achievement asked about, outlining the situation, the task at hand, your action to resolve the task, and the outcomes from your action.

## **Competencies**

*Customer focused:* Build customer empathy, listen to them, understand their needs, commit to those needs, and go above and beyond to deliver them.

- *Experiential:* Explain what customer focus means to you and an event where you helped resolve a customer problem based on their needs.
- *Design/Code:* Does your solution and code solve the needs and requirements the interviewer presented to you? Did you confirm and clarify requirements?



*Adaptability:* React to changes in a positive way and move forward as dates, goals, and requirements change.

- Experiential: An event where requirements or constraints changed and how you adapted to them.
- Design/Code: Expect that the technical question the interviewer asked could possibly have different requirements mid-way.

*Collaboration:* Build relationships with peers and customers and work effectively to deliver a solution.

- Experiential: Describe working relationship with your peers, other teams, and customers.
- Design/Code: Collaborate with your interviewer on solving the technical problem! This is the best way to show collaboration in action.

*Drive for results:* Create plans that foresee obstacles and issues, set reasonable goals and experiences, solicit, and accept feedback, and have a passion for the work.

- Experiential: Situations where you worked to improve a design or implementation and encouraged others to do so as well.
- Design/Code: How you approach optimization (e.g., runtime and space) of coding algorithms is one of the best ways to show this competency during your technical question.

*Influencing for impact:* Articulate priorities and requirements, listen to and understand others, adjust communication to match the audience and be respectful in all these interactions to achieve the desired impact.

- Experiential: Situations where you had to influence others to obtain the right results.
- Design/Code: You should convince the interviewer that your solution, design and code uses the best approaches.

*Judgment:* Identify issues, key stakeholders, scope problems to achievable goals, and make decisions with confidence and conviction to apply them to a positive effect.



- Experiential: Situations where you had to make tough or important decisions, especially without having all the details you needed.
- Design/Code: Not every question you may be asked has optimal solutions – how do you decide what's best to implement? What factors do you use to judge tradeoffs?

### **Preparing for a Technical Interview and Whiteboard Coding Session:**

Your technical interview may comprise:

- Coding (problem solving, logic and maintainability and data structures and algorithms)
- Distributed systems design and databases
- System design (scalability and operational performance)

### ***When interviewing, please keep in mind the following:***

- We are looking for strong computer science fundamentals, data structures, algorithms, object-oriented programming and design concepts.
- You should be able to dig deep into architecture and design on a given problem.
- Choose the object-oriented programming language relative to the role that you are most comfortable with. It is not necessary to code in any particular language.
- Ask clarifying questions before jumping into a solution. Share your thought process when solving a problem. The thought process is equally as important as the solution itself!

### **Cracking the Coding Interview by Gayle Laackman**

Among many websites, this is a well-known book that we would recommend. For each category of a problem (e.g. binary trees, bit arithmetic, logic, graphs, etc)



there is a summary, hints, and tips to look out for when answering the questions, sample questions and answers. Studying the performance/memory/space restrictions when answering the questions would be a stand-out.

### **YouTube Videos**

There is a great series on technical interviews on YouTube by Dickson Tsai, a tutor at UC Berkeley, called [Data Structures in 5 minutes](#). The videos concisely explain topics. The main drawback is that the videos are not shot professionally, but they do convey a lot of information in a short amount of time.

### **Big-O Cheat Sheet**

A [list of all of the algorithm and data structure Big-O complexities](#) in a single page. It includes best, average and worst-case time and space complexity and links to the relevant Wikipedia articles.

### **Websites**

[Glassdoor](#) and [Career Cup](#) are good sources of information about your potential employer, and the kinds of questions that might be asked in the interviews. Do take the reviews with a grain of salt. If you crawl through the reviews and interview question examples, you will get a good idea about the kind of topics they cover, and possibly how difficult the questions might be. Solve as many problems as possible at [leetcode](#) to build your confidence as well.

### **Additional Links**

- <http://bit.ly/NailTheTechInterview>
- <https://www.hiredintech.com/system-design/>
- <http://blog.gainlo.co/index.php/category/system-design-interview-questions/>
- <https://www.educative.io/collection/5668639101419520/5649050225344512?authorName=Design%20Gurus>