

A 9-step plan to approach the software engineer interview at Microsoft and other FAANGs

A former SWE from Microsoft shares his approach

Preparing for interviews for big tech companies can be challenging. Going through rigorous research and understanding the culture is a hefty task. There are a lot of things which are missing at a rate. The more significant problem is that we have a lot of information, but a lot of information is not accurate, as it is not structured right. Hundreds of websites try to tell you how to crack an interview, but none tell you how to be a sound software engineer. In this article, we sit down with Harsh Mittal, currently working with Amazon and a former Software Development Engineer at Microsoft for four years. Through this article, we will discuss a nine-step plan for people who are interviewing at Microsoft or for any tech company. We discuss the differences between interviews at Microsoft and Amazon and the processes involved.

At Microsoft, a plethora of questions in an interview is design questions. The company looks out for more culture-oriented people. They want their employees to be tech-savvy and aligned with their culture. While interviewing, they want to make sure that you believe in the vision of Microsoft and that a candidate is a person who believes in the betterment of the world and wants to be a part of the change. Microsoft appreciates a growth mindset, diversity, and inclusivity. In Amazon, the emphasis is on leadership principles and the importance of everyone being aligned with the principles.

After joining Amazon, one thing that has completely changed for me is the fact that Amazon is an execution engine. They take stuff, they execute and they execute it with the land.

Here's the nine-point plan to approach the software engineer at Microsoft with a coding problem. Even though this plan is now more generic and applies everywhere, you have to ensure that you are not getting lost in interpretation while approaching.

1. **Unclear? ASK-** Ask clarifying questions, as it gives you time to think because if you don't ask clear and intelligent questions, there are chances you might ask unintelligent ones and put yourself in an uncomfortable situation.
2. **Think out loud-** While studying the clover, try to make links in your mind. That is the only time you will get to think about things by yourself when the interviewer is speaking. If the interviewer is not saying, you can communicate your thoughts. As interviewers, they want to see how your brain works the minute a question is given to you. While answering, try to put yourself in that place where you can talk openly about your thoughts or communicate openly. Try to structure how you solve the problem, don't just start coding. The first thing is trying to ask verifying questions.
3. **Draw examples and think of the expected output-** Find the results and verify the results with the interview. That helps eliminate any assumptions you must have made in your head. That strengthens your interpretation. It ensures that you and the interviewer are on the same page about what's about to be done.
4. **Identify the family of patterns-** While interpreting the patterns, memorizing the solutions is probably a bad idea. However, identifying the patterns used across various questions can be helpful during an interview. When you see a question, get your intuition to use it, and remember the family of patterns to which a particular problem may belong. Try to sort of map it to the patterns you have in your head and which pattern it falls for. Dynamic programming has multiple patterns in itself. That's what number five, number six was identified. The data system that you want to use, which data structure do you want to use? It's a good practice to think it out at this stage.
5. **Breakdown problem into primitive steps-** In this step, try to simplify the problem. For example, you have to check if the characters in one sub-string are present in the other and in the same order. Simplify, scale and meet the requirement of your problem.

6. **Identify base cases-** Identify where your base cases are and where the program stop. Understand what the inputs that will have a pre-determined output are.

💡 *Try to note down the stuff you must study for, the topics you must learn about and write the associated source. Take time out, take a day out and write down all the subjects. If you see system design, maybe write on a few YouTube channels from which you want to study it. Or perhaps if you have a mentor or study guide, talk to them and tell them that these are topics that you want to learn about and how creative you plan with them.*

7. **Identify data structure to use-** Focus your solution around how that structure is necessary. While interpreting your case, identify which data structure to use, and walk the interview through your thought process about why you are using a specific data structure. At any point in time, look for the best data structure that reduces time complexity as much as it can.

8. **Implement first. Improvise second-** In an interview, it is natural for you to panic, and the mind tends to wander off. In that situation, implement the first solution that occurs to you and then try to look for improvement. Try and understand how you improvise and optimize it by going through it. Your goal should be to reach an optimal solution by the end of your answer.

9. **DRY-RUN! Bonus: Use edge cases. Example - What if the value entered is negative? Or crosses the maximum value threshold-** Give the interview to the Time and Space Complexity program, so they know what you have done. Make sure that you communicate that. Make sure you talk about the program like you know what you have done, not about something you know what?

It's not always about knowing what you have to do. It's not always about knowing what the solution for a program is. It's about knowing how well, you know, the concepts are involved in this particular question.

💡 *Make notes for yourself and concise messages. Don't just create, like write a paragraph, because you are the one who's going to have to revisit it*

later. Make notes for yourself; you're not making it for anyone else. As long as you understand, it's okay.

To put the matter in a nutshell, Harsh Mittal says, "Make sure you are not getting comfortable, and you will have a lot of cultural learning and best practices. There are so many soft skills that I have developed at Microsoft that I don't think I could have originated anywhere else." As far as interviewing at Microsoft goes, ensure you study the company. At this point, there are plenty of resources available that everyone talks about the Microsoft interview experience. The company wants to ensure that you, as a candidate, are performing well and having a good time. And I think that's for all the companies, no company wants to intimidate you, even at Amazon, or no company wants to intimidate you in the interview process. But one thing that you have to be mindful of is that you know your stuff.