**PREPARATION STRATEGY**

**BEHAVIORAL INTERVEIWING**

I suggest thinking over your past 2 years of experience and listing out 8-10 situations in the past that allow you to showcase [**Amazon’s Leadership Principles**](https://www.amazon.jobs/en/principles). Map out the situations and related data using the points of the STAR Method. Practice answering questions with the STAR method to get in the habit of thinking and talking about experiences in that framework. Candidates have previously shared that [**this article**](https://www.linkedin.com/pulse/3-tips-get-job-amazon-francois-leborgne?trk=eml-email_feed_ecosystem_digest_01-recommended_articles-10-TixCompany&midToken=AQHYJ1Rv8HK7iw&fromEmail=fromEmail&ut=3AAJhvlWv8KnQ1) was helpful for going deeper on this preparation strategy.

**Hint:** Think in terms of highest level scope possible when it comes to your experiences. This can be scope of impact, problem complexity, project complexity, visibility, and/or cross functional collaboration.

**Hint:** Think of some examples about difficult topics and situations like feedback, tradeoffs you’ve made in decisions, disagreements about ideas/decisions, etc… How did you take action to turn those into successes?

**Additional Resource:** [**Leadership Principles Video**](https://youtu.be/CpcxVE5JIX4)

**Question Types:** Most questions will aim to evaluate your problem solving skills especially in ambiguous situations.

* **Previous Experience**: Tell me about a time when…
* **Case**: What would you do in this scenario?

**The STAR Method:**

* Utilize the [**STAR**](https://www.amazon.jobs/en/landing_pages/in-person-interview) method when answering such questions to keep your answers succinct and concise.

**Communication:**

* **Executive Communication:** Index on executive communication. You need to be able to quickly and concisely convey high level details, and be ready to dive into deeper detail in response to follow up questions.
* **Be Specific:** Reference data points that catalyze or justify decisions you’ve made and the results you’ve gotten.
* **Ask Questions:** Ask clarifying questions. Even behavioral questions and follow up questions can have some ambiguity and be open to interpretation. Ask clarifying questions as though the interview needs data points to support hiring you, and it’s your job to help them get the data points they need.
* **Take Ownership:** Be mindful of appropriately using “I” vs. “we” statements. Be specific in identifying exactly what you’ve done vs cross-functional partners, peers, Managers, or customers. Using “we” makes it difficult to understand the impact you’ve had.

**TECHNICAL INTERVEIWING**

The 4 main areas you'll be focused on for the technical side of your interviews will be Logical & Maintainable Coding, Coding (Problem Solving), Data Structures and Algorithms, and System Design. A brief overview on a wider range of topics that will fall into these categories is available [**here**](https://www.amazon.jobs/en/landing_pages/software-development-topics).

**HINT:** ALWAYS ask clarifying questions before solving a problem. We don’t ask trick questions, however, asking questions as though there is something missing that you need to uncover can help you to think more critically and get you to a better solution more quickly.

**HINT:** The interviews are meant to be collaborative. Talk through your process out loud with the interviewers and work to solve the problem with them.

**Coding**

* Be clear in your analysis, problem breakdown, and organization.
* Write clean and clear code. (Syntactically correct – no pseudo code) You'll be assessed on style, correctness, and complexity.
* Complete a working brute-force solution at minimum. The best result is to have a working optimal solution.
* Call out your edge cases. If you have time add unit tests. If you don't have time, at least call out your thought process.
* [Amazon Coding Interview Preparation](https://youtu.be/mjZpZ_wcYFg)

**Data Structures**

* Wikipedia actually is a great resource to brush up on data structures. <https://en.wikipedia.org/wiki/Data_structure>
* There are more in depth tools for brushing up at the bottom of this Wikipedia page under “External Links”.

**Algorithms**

* Wikipedia actually is a great resource to brush up on data structures. <https://en.wikipedia.org/wiki/Algorithm#Classification>
* Know how and when to use breadth-first search versus a depth-first search and what the trade-offs are.
* Traversals and divide and conquer.
* Time complexity.
* Knowing the runtimes, theoretical limitations, and basic implementation strategies of different classes of algorithms.

Go back and re-educate yourself on all data structures and data structure algorithms. They come up all the time. Understand all time and space complexity. Brush up on big O notation. Make sure you even get into hashmaps, binary trees, b-trees (and variants), caches (and associated algorithms). You may want to visit [www.codechef.com](http://www.codechef.com/)

**Systems Design**

* You will be asked to design a software system. Be sure to ask clarifying questions to understand the problem you are being asked to solve and gather requirements.
* Restate what you think you are having to do, state any assumptions and think out loud.
* Expect to start high level and go deeper and deeper into the design.
* You will be asked follow up questions as you are going along on your design. Be clear on the reasons for technical tradeoffs.
* You will want to diagram your system out on the board and then start coding it out.
* Predict how your system will interact with others and how you plan to have yours not disrupt the other current systems in place.

**Systems Design Resources**

This is a list of resources that candidates have previously shared with me that were helpful for them.

* [Amazon System Design Preparation](https://www.youtube.com/watch?v=gNQ9-kgyHfo)
* [Guide to System Design](https://towardsdatascience.com/the-complete-guide-to-the-system-design-interview-ba118f48bdfc)
* [Grokking the System Design Interview](https://www.educative.io/courses/grokking-the-system-design-interview)
* [Gainlo System Design Questions](http://blog.gainlo.co/index.php/category/system-design-interview-questions/)
* [Design a Parking Lot](https://www.youtube.com/watch?v=DSGsa0pu8-k)
* [Design Twitter](https://www.youtube.com/watch?v=KmAyPUv9gOY)
* [Load Balancers](https://www.youtube.com/watch?v=escR-07yVAs)
* [Caching Challenges and Strategies](https://aws.amazon.com/builders-library/caching-challenges-and-strategies/) and [More Caching Strategies](https://docs.aws.amazon.com/AmazonElastiCache/latest/mem-ug/Strategies.html#Strategies.WriteThrough)
* [high scale architecture](http://highscalability.com/blog/category/example)