Onsite Interview

Below is a roadmap I've created for you that has everything you need to help you succeed the Amazon interview process and receive an offer. **Preparation is absolutely essential and please use this as a checklist to prepare for the interview**.

First steps first :

1. Before jumping into any code, answering any questions, etc. – **ASK CLARIFYING QUESTIONS**. Make sure you are fully aware of what they want to see by asking questions to get more data as needed. **They want to see you dive deep into what they are looking for.** You will be purposefully asked ambiguous questions throughout the interview to gauge your problem solving skills.

2. Code in the language you are most comfortable using.

3. As you are putting your code on the board **THINK OUTLOUD and walk them through the steps you are taking.** It is important to show them your thought process as you answer their questions. Engage them in what you are doing and why you are doing it.

4. They will be gauging your ability to take coaching, come up with alternate solutions, and show your ability to analyze a problem and come up with various solutions.

Interview Format:

1º Interview: Behavioral Questions + Coding & Algorithms & Data Structures

2º Interview: Behavioral Questions + Coding & Problem Solving

3º Interview: Behavioral Questions + Problem Solving & Design (logical & maintainable/software/Object Oriented Design)

4º Interview: Behavioral Questions + Architecture & System Design – This interview will test your understanding of distributed systems, concurrency/multi-threading, scalability, performance, operational excellence, etc. Asking clarifying questions and gathering requirements is KEY for this interview. Please use this link to prepare https://www.hackerrank.com/domains/distributed-systems/client-server/page:1

Onsite Interview

SECTION 1 -
Amazon Leadership principles – click here

The principles are an important aspect of who we are at Amazon and the culture we maintain.

As you review the 14 principles be prepared to answer a questions about your work, delivering on a project, challenges you might have faced, problems you had to work through, meeting a deadline, etc. You will be expected to walk us through various example in your life in which you exhibited the leadership principles.

Ideas when preparing for the Leadership Principles:

Ownership: Think about situations where people said this is not my job and what did you need to do on that situation?

Customer obsession: Think about situations where you went out of your way to satisfy your customer.

Bias for action (goes together with deliver results): Think about times where you had to make decisions quickly by taking very calculated risks especially when there was a lack of data.

Have a backbone, Disagree and commit: Think about times where you were able to push back on your manger or team members by presenting counter arguments with the help of relevant data or metrics. Think about how you influenced your manager or team members to adopt your approach.

STAR methodology

You must answer the Leadership questions using the STAR methodology

Situation: Open with a brief description of the Situation and context of the story (who, what, where, when, how).

Task: Explain the Task you had to complete highlighting any specific challenges or constraint (e.g., deadlines, costs, other issues).

Action: Describe the specific Actions that you took to complete the task. These should highlight desirable traits without needing to state them (i.e., initiative, intelligence, dedication, leadership, understanding, etc.)

Result: Close with the result of your efforts. Include figures or metrics to quantify the result if possible. Provide measurable specifics where you can, and be prepared to back them up.

Onsite Interview

SECTION 2 –

Coding Interview Tips

Your goal must be to provide the interviewer with the **most optimal solution**

Ask clarifying questions to understand the requirements before you start to code

You will be asked to **determine the efficiency and runtime (time/space) complexity** for all coding questions.

Aim to **optimize** the efficiency of your code to make it run faster.

You must be able to code fast but at the same time it **must be clean, clear and concise**.

Be ready to explain how its built-in structures and operations work under the hood, in case your interviewer is not very familiar with the language of choice.

Be especially careful with concise built-in expressions that might have a linear cost underneath (such as list.delete(element) or list.search(element)'.

Don't worry too much about a perfect, fully compliable syntax, **but the code should be syntactically correct** for the most part. If you don't recall a particular syntax construct, **be clear with the interviewer about the assumptions** you're making.

Use proper variable naming functions.

Interviewers do not consider pseudo code as a complete solution to a problem.

Account for all edge cases and corner cases.

Test code in your mind if possible.

PROBLEM SOLVING:

This competency measures the ability to take something complex, identify the problem, break-it-down, and solve it.

The interviewer will share with you an ambiguous problem statement. It may be more difficult to understand what the problem is and **will need to ask clarifying questions to understand how they can start solving the problem.**

Divide the problem into smaller pieces in order to solve it.

LOGICAL AND MAINTANABLE:

This competency measures the ability to write maintainable, readable, reusable, and understandable code.

You'll have to perform a component design and object oriented design properly.

Must review: Inheritance / classes

Divide the problem into small pieces, small pieces of code, instead of a messy function

.

Onsite Interview

Data Structures and Algorithms Preparation:

The interviewer will ask you to solve a data structure oriented problem. You'll then write a code using the appropriate data structure to solve the problem.

You must be able to explain the inner workings of common data structures and be able to compare and contrast their usage in various applications.

You must be able to solve problems in Linear time.

Practice questions on Hacker Rank: https://www.hackerrank.com/interview/interview-preparation-kit

YouTube Link for Data Structures, Algorithms and CS Concepts in general: https://www.youtube.com/user/mycodeschool

Absolute Must Haves :

1. Hash Tables – you must be able to explain how internals of hash tables work for example hashing.
2. Link lists

3. Trees (especially Binary Search Trees) - be ready to explain how a binary search tree works a) if it's balanced and b) if it's not balanced. (use this link to help review: http://cslibrary.stanford.edu/110/BinaryTrees.html)

4. Big O Notation - you will be asked to determine time/space complexity in almost all your interview questions, and how to optimize your code for better time/space complexity. You are extremely likely to see a question where the solution will involve the use of hash tables.

5.Algorithms: Breadth First Search/Depth First Search, Binary Search, Merge Sort and Quick Sort, Sorting

6. Arrays, Recursion, Stacks/Queues, Bit Manipulation, Traversals, Divide and conquer

7. TO REVIEW: Binary Search Tree (++), Maps (hash map, dictionary or hash table) (++), Array (+), Link lists (+), queue, stack, vector and graph.

8. Join two data structures together

Onsite Interview

SECTION 3 –
System Design and Component Design

Component Design:

Review OOD concepts. You can refer to this doc for the review

Object Oriented Design – Very important for Junior level candidates

This video simulates well the Component Design interview

Be ready to define classes, attributes, abstraction, methods and the relationship between components during the exercise

Make sure to explain the trade-offs and the choices you make during the exercise

System Design:

System Design Introduction for Interview (and other great videos from Tushar Roy

System Design Primer

Netflix System design - YouTube

System Design Interview Question: DESIGN A PARKING LOT - asked at Google, Facebook - YouTube

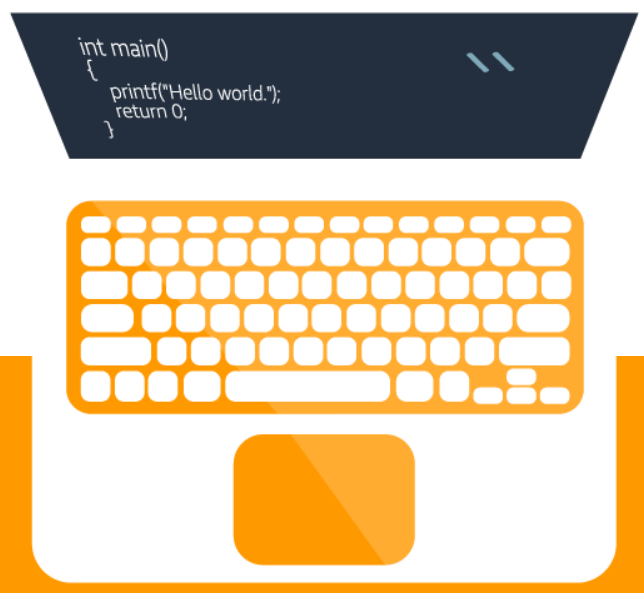
Extra links for System Design:

https://www.youtube.com/watch?v=CtmBGH8MkX4&t=40s

https://www.youtube.com/watch?v=qYhRvH9tJKw

https://www.youtube.com/watch?v=tndzLznxq40&t=2s

https://www.youtube.com/watch?v=tndzLznxq40&t=1s



GENERAL ADVICE:

Be Specific: Reference data points or metrics that justify decisions you've made.

Ask Questions: Ask clarifying questions. There are often ambiguous situations to deal with in the work place, and so there will be a few in your interviews. Asking questions will give you a clearer understanding of the challenges presented by the team, and gives the interviewers insight into how you dive deep on issues and solve problems. This can make a huge difference in you interviews.

Be Honest: Some interview questions may be negatively oriented. (ex. "Can you tell me about a time when you didn't meet a deadline?") We all have setbacks, but what's important is how you handle them, and what you learn from them.

Take Ownership: Be mindful of appropriately using "I" vs. "we" statements. Spend more of your time speaking to your responsibilities and what you've accomplished, as oppose to what your team as a whole accomplished.