Interview Prep

Software Development Engineer

What does a SDE do at Amazon?

Amazon is a place where builders can build. We offer an environment in which employees can invent and innovate on behalf of our customers. We want employees who will help share and shape our mission to be Earth's most customer-centric company. Our engineers tackle some of the most complex challenges in large-scale computing with ingenuity and simplicity. They own the software development life cycle; designing, coding, testing, implementing, maintaining, and iterating on solutions. With a strong understanding and applicable knowledge of CS fundamentals, things like algorithms and data structures are innate to them.

Our evolution from website to e-commerce partner to development platform is driven by the spirit of invention that is part of our DNA. We're making history, and the good news is that we've only just begun.

Want to become a Software Development Engineer at Amazon?

The next few pages will walk through some helpful tips for the interview process.

amazon

Behavioral Based Questioning

What to expect:

Each interview will include 2-3 behavioral based questions. These questions ask about past situations or challenges you've faced and how you've handled them, using Leadership Principles to guide the discussion.

Keep in mind, Amazon is a data-driven company. When answering your questions, your focus should be on the question asked. Ensure your answer is well-structured and provide examples using metrics or data when applicable. Reference recent situations (within the last 2 years) whenever possible.

During your onsite interview, you can expect to have 8 different Leadership Principal related behavioral questions.

When answering these questions, utilize the **STAR method:**

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

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

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

Result: Close with the result of your efforts. When possible, include data to quantify the result.

Leadership Principles

We use our **Leadership Principles** every day, whether we're discussing ideas for new projects or deciding on the best approach to solving a problem. We rely on them to help us make hiring decisions.

The best way to prepare for your interview is to practice using the STAR method (also referred to as SBI) to outline how you've applied using the below Principles in your professional experience. Follow the link above for a detailed explanation of each Leadership Principle.

Take a few minutes to watch the following video:
Leadership Principles & Behavioral Interviewing (6:18)

Deliver Results
Customer Obsession
Have Backbone; Disagree & Commit
Learn & Be Curious
Bias for Action
Earn Trust
Dive Deep
Ownership
Hire & Develop the Best
Insist on the Highest Standards
Invent & Simplify
Are Right, a Lot
Think Big
Frugality

Coding

What to expect:

You will have 3 separate interviews evaluating 3 areas of coding; logical and maintainable, problem solving, and data structures and algorithms.

Your interviewer will pose an intentionally vague question to evaluate your ability/approach to solve problems. Engage with your interviewer and ask clarifying questions.

Consider runtimes for common operations and understand how they use memory. Understand data structures you encounter in core libraries. i.e. trees, hash maps, lists, arrays, queues, etc.

Understand common algorithms. i.e. traversals, divide and conquer, when to use breadth first vs. depth first recursion. Discuss runtimes, theoretical limitations, and basic implementation strategies for different cases of algorithms.

After you've gathered qualifying requirements - translate into clean written code, checking edge cases. While you're coding, ask yourself "If someone looked at this code in the future, would they be able to understand it without explanation, comments, or documentation?"

We expect you to produce enough code for us to evaluate how your solution addresses the problem.

Ensure your code is scalable, robust, and well-tested.

Be familiar with prominent languages, including the syntax of the language.

Be prepared to discuss technologies listed on your resume. i.e. if you list Java or Python, expect technical questions about your experiences with these technologies.

Before Coding:
- Ask clarifying questions
- Voice your assumptions
- Ask for examples and use them to validate your ideas

During Coding:
- Explain your decision making
- Describe your code
- Write syntactically correct code, no pseudo code

After Coding:
- Test your code

Topics to review:

Visit **codechef.com** or similar websites to brush up on problem solving and core CS fundamentals.

The book "**Cracking the Coding Interview**" has a lot of valuable content.

Take a few minutes to watch the following video:
Coding (16:15)

LEARN MORE

Dive into our Leadership Principles
Interviewing for a tech role? Explore our common technical topics
Explore Interviewing at Amazon for FAQs, prep guides and more

QUESTIONS? REACH OUT TO YOUR RECRUITING POC
AMAZON IS AN EQUAL OPPORTUNITY EMPLOYER

System Design

What to expect:

You will have one interview focused on system design. Be prepared to white board. Practice drawing your system design by hand.

Engage with your interviewer and ask clarifying questions. While the interviewer won't try to trick you, they might be intentionally vague to push your innovation. It's important to know what sort of design the interviewer is looking for, so ask questions.

Start with the customer in mind. Who is the customer and what problem are you solving for them?

As you ask your questions, begin writing a list of requirements on the board. This should typically be the first thing you add to the white board. Think out loud as you write out your design and show us your ability to solve problems.

Begin drawing a diagram once you've asked enough clarifying questions and gathered the necessary requirements to begin white boarding your system design solution. Start with shapes to represent different software components and data sources, and then arrows connecting them to show web services, APIs and interactions between components.

Scaling is a critical component of software design at Amazon. It's important to consider scaling when diagramming and designing your software system. Be sure and research scalability concepts and technology prior to your interview such as caching, load balancing, non-relational databases, micro services and sharding.

Know how your solution solves the problem. If you suggest technology to help solve, understand how that technology works.

Operational performance of your design is important. How will you ensure this system is working at an acceptable level of performance? If a problem occurs, what will be involved to trouble shoot and resolve quickly? What are the possible points of failure and how can they be made more robust against failure?

Be prepared to discuss trade-offs in your design. With any software system there are multiple ways to design it. What advantages & disadvantages would yours have? What if you were to change a component or process?

Topics to Review:

Often times, software systems need software components, something to store data, something to make decisions (such as business logic) and APIs or processes. Reviewing software systems design diagrams (especially SOA or distributed software systems) can be helpful for preparation

Knowledge of distributed systems, SOA, and n-tiered software architecture is very important in answering systems design questions. If you don't work with these concepts regularly, be sure to review them.

Take a few minutes to watch the following video covering Design: **System Design** (10:42)