INTERVIEW PREP

Software development engineer

WHAT DOES AN SDE DO AT AMAZON?

Well, we tackle some of the most complex challenges in large-scale computing. We own the software development life cycle, design solutions and work on coding, testing, implementing, maintaining and iterating solutions. With a strong understanding and applicable knowledge of CS fundamentals, things like algorithms and data structures are innate to us.

Want to become an Amazon software development engineer? Let's walk through some helpful tips for the interview process.

SDE tips

Be familiar with prominent languages, including the syntax of the language. Ultimately, pick the one you're most comfortable with and stick with it.

Showcase your knowledge of system design. i.e. basic distributed computing concepts to help formulate answers to more complicated distributed architecture questions.

Understand the inner workings of common data structures. Be able to compare and contrast their usage in various applications.

Research scalability concepts and technology. i.e. caching, load balancing, non-relational databases, mirco services and sharding.

Write syntactically correct code—no pseudo code. Ensure it's scalable, robust, and well-tested.

Use object-oriented design (logical and maintainable code) best practices to build lasting software.

Be familiar with OS topics such as memory management, processes, threads, synchronization, paging, and multithreading.

Highlight your knowledge of software systems design with more emphasis on low level design. High level design is encouraged, but not required.

Be able to design a system at a high level. Be comfortable diving deep into specific areas.

SDE technical tips

Be familiar and ready to solve in-depth technical questions on concepts like data structures and algorithms. This will likely include qualifying requirements, checking edge cases and white boarding your solutions with our engineers. In-person interviews tend to be more in-depth than the types of questions asked during your phone interview.

Be prepared to discuss technologies listed on your resume. i.e. if you list Java or Python as technical competencies, expect technical questions about your experiences with these technologies. It is helpful to review the job description before your interview to align your qualifications against the job's specific requirements and responsibilities.

Visit codechef.com or similar websites to brush up on problem solving and core CS fundamentals. "Cracking the Coding Interview" is a good read.

White boarding

Be prepared to white board. Practice writing code and system design out by hand. Consider a scalable code or design before you begin drafting.

Interact with your interviewer. You will be asked several questions related to design. Engage with your interviewer with necessary questions to complete the exercise.

Dig for clarification. Your interviewer will not try to trick you. Questions may be intentionally vague to push your innovation.

Begin drawing a diagram once you've done enough digging 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.

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

Think out loud as you write out your code or system design.

Show us your ability to solve problems.

Scaling is a critical component of software design. Consider software components, how to store data, APIs, component relationships, and data flows. Keep this in mind when diagramming and designing your software systems.

Be familiar with Distributed Systems, SOA, and n-tiered software architecture. This will help you answer system design questions.

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?

Keep the customer front of mind. Who is the customer, and what problem are you solving for them?

Write a list of requirements on the board, and keep asking questions. This should be the first thing you write out.

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