SDE Screening and Prep material.

Coding:

1. Discuss the approach with the interviewer before writing the solution on whiteboard.
2. Remove the ambiguity first before coming up with a solution.
3. No pseudo codes, write the full code in the specified language
4. Run time complexity discussion and optimization
5. Logical maintainable code design using Object Oriented Programming principles
6. Language specific data structures, templates, generics, classes -> Java or C++
7. Write comments with the code and make sure you use right variable names.
8. Test cases and edge cases discussion
9. Memory optimization in the code, not using extra space if not needed.

Design:

1. Discuss the use cases of the application, audience that will receive the service.
2. Discuss whether the service is read heavy or write heavy. (For e.g. designing an FB newsfeed is read heavy as more people read the feed than posting something on Fb.)
3. Discuss the scalability with the interviewer if the number of audience is small say 10,000 or a million users will use the service.
4. Come up with an overall block design diagram comprising of layers in the demand as the web browser, server, cache, database etc.
5. The flow of the traffic must be clearly shown in the diagram.
6. Discuss the storage model as in the data will be stored in a NoSQL database or a SQL database is good choice in the situation.
7. The differences must be discussed and after the comparative analysis, one choice must be made as in NoSQL or SQL.
8. Optimizations must be discussed with the interviewer as in what can be done if the traffic grows and how to return to the top consumers of the service.
9. Concurrency, distributed systems and cache memory management are often discussed in optimizations and must be clearly discussed with the interviewer.
10. The interview might even ask for a low-level design of the application, so in that case classes, interfaces and other OOP details are needed to be answered.

Screening questions-

1) Tell me about a large Software project (has the most users/transactions) that you have designed architecture for (or designed parts of it)?

   Involvement

   - How did you get the design started? How was it organized? Were you a part of the high level design?

   □Looking into the candidate’s responsibilities in design: gathering requirements, making design decisions, influencing design decision, owning features

   - How many users? Or how many transactions?

☐ Looking for a big number, thousands, millions, as close to Amazon as possible

Design

- What was the tech stack?

☐Object Oriented programing languages. Looking for Java, C#, C++, Python, Ruby, Groovy

- Describe the system you designed.

☐Looking for Service Oriented Architecture, Distributed System, SAAS (Software as a Service), PAAS (Platform as a Service)

☐Why were the architecture choices decided that way. What were the tradeoffs?

Scaling

- What did you do to make sure the system is scalable?

☐Looking for sharding, caching, load balancing, adding micro services, using non-relational databases

- What were some of the challenges/bottlenecks you encountered when scaling up the system?

☐Challenges? Looking for uptime/crashes, bugs, latency, reliability, availability

- What did you do to ensure the application is meeting its goal and delighting the users?

☐Looking for debugging, patching, survey, feedback, A/B testing

- What did you do to ensure continuous improvements?

☐Quality Assurance: white box testing, smoke testing, work with QA team closely to monitor performance

2) How do you implement reusability in code or modularity? For example- if as an Amazon engineer I want to get a notification as soon as someone cuts a ticket (severity 1-6) and way the code is written it only supports Sev 1 to Sev 5, in case I want to add Sev 6 and my code doesn’t support this functionality it means there was a major issue in code maintainability and extensibility.

3) Can you elaborate about your current project, what type of components are you working on, tech stake being used, what design pattern are you using? For eg- An experienced engineer would say he used high level design for the components, also talks about storage layer, DBS, etc (dynamo db aws). Any engineer working around large scale data would use Hadoop, Map reduce and for data analysis he might use Elastic search. For messaging he might be using pub/sub model AWS SQS, for data streaming he may be working on Kinesis (if someone express interest in prime videos).