

Mobile Developer Interview Preparation Document

Machine Coding

You will be given a small app to implement. You are expected to submit a working solution in given time (typically 90 minutes).

Expectations

- Demoable code
- Clear Separation of Concerns (Good Class Design)
- Functional Correctness and Completeness
- Readability
- Modularity and Extensibility
- Exception handling

Problem Solving and Data Structures

This round will focus on your problem solving skills and knowledge of common data-structures. You will be given a problem statement (real-world or algorithmic) and you will need to provide a solution in the form of pseudo-code.

Expectations

- Ability to identify problem type and apply conventionally known techniques to solve them.
- Usage of optimal Data Structures for the problem statement
- Good understanding of Time and Space complexity
- Identify edge cases and boundary conditions of the problem
- Ability to test the correctness of solution

Domain knowledge (Android) --- only for Android candidates

The purpose of this round is to understand your knowledge and understanding of Android.

Expectations

- Write something here
- Understanding of Android Pillars like Activity, Service, Content Providers, Broadcast Receiver.
- Good understanding of Android Architecture.
- Multithreading concepts.
- Lifecycle of Android components. When to use which component.
- DB handling.
- Performance measurement & debugging.
- Usage of ADB command.
- Network Understanding and good to have tool knowledge like Charles.
- Testing Frameworks.

System Design round

Given a well-defined problem statement, you will need to convert Product Requirements into a working design. This will be a very open-ended conversation about how we build complex features or design entirely new products from scratch.

Expectations:

- Ability to design and plan a product/feature independently
- Prioritisation of features/tasks
- System design and architecture for this product's implementation
- Design APIs and/or interfaces for the various pieces
- Design network layer for the problem: Decide when to make what calls
- Design database layer for the problem with right schema
- Use performant UI components that fit the use-case
- Ability to go deep into some features and look at factors like product decisions, limitations etc amongst many others