Interview questions:

1) Introduction and android projects experience?

3) Architectures used in Android(like MVP and MVVM)?

4) Describe MVVM and MVP?

5) How is MVVM different from MVP?

6) Explain what is meant by LiveData is lifecycle aware?

7) Live Data can be put only in View Model or Any where else like Services or Presenter etc.?

8) What is unit testing and UI testing? For what (files) will you have to write unit test?

9) Explain the basic flow of hitting weather API in MVP or in MVVM? (Since I had mentioned weather API that I used in a project)

10) How do add header to a post  request in Retrofit?

11) What is the advantage of using Retrofit over other networking libraries like Volley?

12) What is data binding? What are the advantages of using it?

13) What are Co-routines in kotlin?

14) Question based on design.(A UI mock up/design was shown how will you design the layout file for it)

15) Kotlin – sealed classes, extension functions, scope functions, lambda functions

16) Is there ternary operator in Kotlin?

17) What is a component in Dagger?

18) What ‘provides’ do in Dagger?

19) Git, code merging, PRs, branch creation strategy during sprints and releases

20) How to resolve circular dependency in Dagger?

21) Activity, Fragments lifecycle and launch modes

1. What are the libraries, third party SDKs and architecture that you used in your previous project?

2. How the MVVM handle the memory leaks?

3. In which scenario, the memory leaks are happen?

4. How you find the memory leak is happening?

5. How to avoid the memory leaks?

6. How the live data works?

7. Difference between observe and observeForver in livedata

8. What is ANR and asked about multiple scenarios that ANR happen

9. How to avoid the ANR

10. What are the jetpack components that you used in project and purpose of the components

11. When the system throws Out of Memory exception and how it will be handled?

12. Why we are using Dependency injection

13. What does the @provides do in Dagger2

14. What is coroutine?

15. What does the suspend function do in coroutine?

16. Before the activity is created, I put the app to background and takes to foreground, What is the view model lifecycle called? And still the background thread runs and get the results from API?

17. Asked questions from previous project experience

New questions:

Implementation of navigation graph  
Scenario explanation with navigation usage between fragments  
Difference between navigation and fragment implementation with intent  
Main functionality navigation graph  
How to parse data between fragements in navigation graph  
Difference between MVVM and MVP  
How to know current destination between fragments in navigation  
Live data  
Mutable livedata vs Mediator livedata  
How to observe both mediator live data  
How to clear observer in viewmodel  
Difference between Android view model and view model  
Why View Model has preferrable over android view model  
How to change store live data before observe  
Corountines concept with scenario explanation  
Dispatcher Main vs IO  
Why Dispatcher IO preferrable in background thread  
What are concept do you used in your experience to perform background thread?  
Difference between service vs work manager vs coroutines  
Main thread vs background thread  
Async task disadvantages  
Device orientation scenario issue and how to handle it  
Any two challenges faced in your experience and how you handle

 RxJava:

1. Reactive programming concepts
2. RxJava - Observables, Observer, Schedulers
3. Operators like map, flatmap, concatMap, switchMap, zip, debounce
4. Flowables and backpressure strategy

Kotlin topics:

* Coroutines, Structured concurrency
* Lambda functions
* Scoping functions (let, apply, also, with, run)
* Higher order functions,
* Extension functions

Non-technical questions:

1. Introduction about your work experience
2. How you handled the clients and team
3. What and all the metrics you followed for code governance
4. how you handle the status report
5. what and all the issues you have faced in that and how you overcame that