1. how much exp with kotlin
2. Disadvantage of asynctask,
3. Benefit of asynctask, retrofit (what , when , why)
4. Major diff between mvc and mvvm design pattern (why we use them)
5. Have u work on dagger dependency, hilt
6. Services, intent services and work manager (what , when , why)
7. Launch mode (what call back on single top).
8. What type of challenge face on current project.
9. Fragment life cycle and when we come from fra2 to frag1 , what life cycle will call. For both.

**HCL:**

activity life cycle, ++done

architecture, +done

fragments… +done

architecture components,+done

Ans: Linex carnel — all the driver class

Library - openGL, sqlite - Android Runtime. -core library

Application framework - all manger class - like location manager, telephony manager

Application : = contact, email , media player

Services, +done

Broadcast receivers ++done

Threading concept,

Java questions on OOPS,

how you pass data between fragment,

fragment lifecycle, +done

difference b/t listview and recycle view, +done

methods in adaper for recycle view ++done

What is the difference between a process and a thread

**Ans:** Process: - a program which is in exciting state is call called process

thread:- is a subpart of subprocess. It is light weight

Context switching takes more time in process

Thread tekes less time

asked to write program prime number and palindrome.

String Reversal,Find third largest number in an array

* **[1) What is Singalton design pattern?](https://www.onlineinterviewquestions.com/hcl-android-developer-interview-questions/#question1)**
* **[2) What are Activities in Android?](https://www.onlineinterviewquestions.com/hcl-android-developer-interview-questions/#question2)**
* **[3) What is the difference between a process and a thread?](https://www.onlineinterviewquestions.com/hcl-android-developer-interview-questions/#question3)**

A process is an instance of a program that is being executed.

The process and thread are an independent sequence of execution, but both are differentiated in a way that processes execute in different memory spaces, whereas threads of the same process execute in shared memory space.

* **[4) What are Broadcast receivers in Android?](https://www.onlineinterviewquestions.com/hcl-android-developer-interview-questions/#question4)**
* **[5) What is Android ANR?](https://www.onlineinterviewquestions.com/hcl-android-developer-interview-questions/#question5)**

**Application Not Responding** (ANR) errors are triggered when the UI thread of the application is not responding for more than 5 seconds

* **[6) What is a Fragment?](https://www.onlineinterviewquestions.com/hcl-android-developer-interview-questions/#question6)**

A fragment is **a self-contained, modular section of an application's user interface**

* **[7) What are Intents and services in Android?](https://www.onlineinterviewquestions.com/hcl-android-developer-interview-questions/#question7)**
* **[8) What is Orientation in Android?](https://www.onlineinterviewquestions.com/hcl-android-developer-interview-questions/#question8)**
* **[9) List out different states of an activity?](https://www.onlineinterviewquestions.com/hcl-android-developer-interview-questions/#question9)**
* ++ Activity life cycle
* **[10) What are different data types supported by AIDL?](https://www.onlineinterviewquestions.com/hcl-android-developer-interview-questions/#question10)**
* **[11) What are Dalvik Executable files?](https://www.onlineinterviewquestions.com/hcl-android-developer-interview-questions/#question11)**
* **[12) What is Android Emulator?](https://www.onlineinterviewquestions.com/hcl-android-developer-interview-questions/#question12)**
* **[13) What is Context in Android?](https://www.onlineinterviewquestions.com/hcl-android-developer-interview-questions/#question13)**

Oops concept

How to make a class pure immutable

String class immutable explain in deep

Exception and error

Hashmat vs hashtable

Abstraction use in project

Diff between Throw and Throws and Throwable

Types of exception

What is scope in coroutine

Lateinit and lazy write code

Find max element from array and its complexity

The throw and throws is the concept of exception handling where the throw keyword throw the exception explicitly from a method or a block of code whereas the throws keyword is used in signature of the method.

Class test{

if(amount < 500)

Throw new ArithmeticExcaption(“amount is less”)

}

Class Test {

Public static int devidenum(int a, int b) throws Arithmetic exception

Int Num = a/b;

Return num;

}

**MobilePrograming**

**SET 1**

1. Launch Modes

2. Live Data

3. Dagger

4. Data Binding

5. Weak Reference

6. Difference between arraylist and linkedlist ++done

7. Difference between array and arraylist. ++done

8. Linked list  ++done

9. Finding element using arraylist and linkedlist

10. ThreadpoolExecutor

11. Coroutines

12. Pending intent ++done

13. Loading thousands of images in RecyclerView

14. Pagination Library

15. Navigation in Android ++done

16 . MVVM

17. Handler in Android

18 . Builder pattern

19. Difference between string builder and string buffer ++done

20. Threads

21 . Intent service ++done

22. Difference between lateinit and lazy kotlin

23. WorkManager. ++done

24. Mutable livedata ++done

25. Viewmodel ++done

26. Memory leakage in Android ++done

27. Message Queue

28. Difference between serializable and parceble ++done

29. Job schedulers

30. Android clean architecture

**SET 2**

1. Different launch modes in android.  Detailed

2. Activity life-cycle and fragment life-cycle.  Detailed

3. Service life cycle. What are bound services. How to handle background tasks above Android-N.

4 Communication between activity  and  services.  Detailed

5. What are task affinity.

6. Difference between listview and recycler view.

7. Importance of ViewHolder in recycler view.

8. How to download the file once the downloading stops in between

9. Data binding in android.

10. View model in android.

11. What is gradle and its uses.

12. Steps to create an apk file.

13. What is proguard and its uses.

14. What is jetpack.

15. Cancel the api request in retrofit.

16. What is Dagger and its uses?

17. How to improve the code quality.

18. What are handlers in android?

19. Difference between thread and service.

20. Design patterns- Factory pattern, build pattern, singleton pattern.

Tell me about ur self ?

1.What is collection ?

2.What is component ?

3.What is MVVM ?

4.What is MVC ?

5.What is a hashmap and advantage.

**Set 3**

Questions:

1) inheritance

2) polymorphism

3) Array List vs Linked List

4) stack and queue

5) service and intent service

6) Service types

7) collections and its frameworks

8) fragment lifecycle

9) ways of passing data between the activities.

10) MVVM

11) thread vs coroutines

12) Launch modes in android

13) sticky intent

**Set 4**

1.What is collection ?

2.What is component ?

3.What is MVVM ?

4.What is MVC ?

5.What is hashmap and advantage

**Adobe**

* Certificate Pinning

Kotline:

Coroutine

Scope functions

● Difference between livedata and flow.

* Difference between lazy and lateinit.

Sealed Class

Intent Service - Callback method

Rxjava related questions.

Data structure questions with its use cases like when we use

Design patterns.

**Question:- .diff b/w comapare and comparator**

**Ans:** -  Comparable provides a **single sorting sequence**. In other words, we can sort the collection on the basis of a single element such as id, name, and price.

The Comparator provides **multiple sorting sequences**. In other words, we can sort the collection on the basis of multiple elements such as id, name, and price etc.

Question: -what hashcode method will do

Ans:- hashCode in Java is a function that **returns the hashcode value of an object on calling**. It returns an integer or a 4 bytes value which is generated by the hashing algorithm.

Question: - Enum:-

Ans:- in case of Enum we restrict the value.

Like we have set of **Value** of play with that value.

sealed classes:- But in case of sealed we restrict the **type** of those value we can use only.

Like we have set of **type** of play with that object. This is abstract class we can not create object

**[Sealed classes](https://kotlinlang.org/docs/reference/sealed-classes.html)**are used for representing restricted class hierarchies

Question: Lunch mode

Ans: Launch mode is an Android OS command that determines how the activity should be started.

1: Standerd

2: single top

3:single task

4: single instance

Question:- what is coroutine

Ans:- **coroutines are lightweight threads**, they **use** **predefined thread pools and smart scheduling for the purpose of which task to execute next and which tasks later.**

There are mainly two types of functions in Kotlin to start a coroutine:

* launch{ }
* async{ }

**Launch function:-** The launch will not block the main thread. If users do not want to use the returned result.

**Async:-** The Async block the main thread. If users want to use the returned result

Question: run time polymorphism

Ans:- when an object is bound with functionality at runtime this is know as runtime polymorphism.

Can be achieved by method overriding.

Question: what is lama expression

Ans:A **lambda expression** is a shorter way of describing a function. It doesn’t need a name or a return statement.

Exp:- val lambda1: () -> Unit = { println("Hello, world") }

Question: MVVM why we use

And: Main thing is 1) separation of concern it means there is no affect any layer

2) unit testing

Question:Work manger[](https://www.youtube.com/watch?v=kT4tT-55Gpw)

Ans:

Question: what is null sefty.

Ans. Kotlin null safety is **a procedure to eliminate the risk of null reference from the code**. Kotlin compiler throws NullPointerException immediately if it found any null argument is passed without executing any other statements.

Question: lateinit Vs lazy

Ans: **lateinit can only be used with a var property whereas lazy will always be used with val property**. A lateinit property can be reinitialised again and again as per the use whereas the lazy property can only be initialised once.

Question: what is scope fuctions.

Ans:- scope function purpose is to execute a block of code within the context of an object.

Types of scope functions:-

let, run, with, apply, and also.

Let- perform multiple operation on object the we use let. (It)

Apply - point to object exp- web view have diff-2 property so we can use. (It)

With- behave like object - whatever u pass object it work according to this object. (This)

Run- same behave like object - whatever u pass object it work according to this object. (This)

It is combination of with and let.

also- same as let, perform some operation on object (it)

**Fun main() {**

**Val amp = Employee();**

**Amp.name = “xyz”**

**Amp.age = 20;**

**Amp.let {**

**println(it.name)**

**println(it.age)**

**}**

**}**

**Exp =**

**with(emp) {**

**age= 30**

**Name: “xyz”**

**}**

**Question:-** Extension functions in kotlin.

Ans: Kotlin provides the ability to extend a class with new functionality without having to inherit from the class.

This is accessible by library class and user defined class

For example, you can write new functions for a class from a third-party library that you can't modify. Such functions can be called in the usual way, as if they were methods of the original class. This mechanism is called an *extension function*.

Exp: fun String.findChar(intdex:Int):Char = this[index]. (Exp of library class)

Fun main {

System.out.println(“xyz”.findChar(2))

}

**Another is example in android make a toast as extension function:**

Like- Context.toast(msg:String){

Toast.makeToast(this,””,Toast.lenghShort).show()

}

And Call ->

toast(“hello”)

**User Defined Extension:-**

class someExample {

companionObject {

}

}

Fun someExample.companion.someexample(){

println(“extension function”)

}

Fun main{

someExample.someExample()

}

Question: what is paging

And: It is part of android architecture. Which is help you load and display small chunk of data.

**Question:-** Difference between setValue() and postValue() in Kotlin MVVM

**Ans:**

While using the Main thread to change the data, you should use the setValue() method of the MutableLiveData clas s and while using the background thread to change the LiveData, you should use the postValue() method of the MutableLiveData class.

Write code singleton . And its useCase

Val vs const, use case

Lateinit vs lazy where we use

Coroutine in deep, why we uses suspend function - some question

Multiple execution- api call

Mvc and mvvm why we use - scenario based question on mvvm part

Screen orientation how would u manage- when we use

Life cycle of activity when we rotate portrait to landscape

**Android developer with kotlin :**

**Questions Asked in Interview ThoughtWorks**

1. Project-related questions suggest some alternate way of doing the same which u have done earlier in those projects.
2. Mvvm related questions(describe each component you use to achieve this).
3. Dagger-related questions(Annotations).
4. Multi-module architecture.
5. Make your own architecture without using any jetpack component which manages the lifecycle.
6. Webrtc related questions how we can share the screen how we give access overcall.
7. Threading how you achieve that in kotlin or in general if you have too many threads running how will you manage them
8. Question related to coroutines.
9. Synchronized keyword
10. How do you prevent from being another instance of the same class?
11. Question-related to the problem statement that you have 7 boxes in those boxes one is heavier. You need to find out the heavier one and tell me the number of iteration you need to achieve this.
12. Questions related to the parallel execution of threads.
13. Navigation components of the jetpack.
14. Paging
15. Database migration-related question

**Question asked in an interview: Reliance**

**Summary:** The whole interview went on the use cases and scenarios-based questions related to the components.

1. Introduction and then questions start from the projects you have done.
2. RecyclerView handling with different layout managers.
3. The difference between listview and recycler view.
4. recycler view internal working.
5. Design patterns.
6. MVVM-related questions.
7. Jetpack component with detailed explanation.
8. How to handle the socket and how to deal with audio-video calls because I have done a project on that related to that particular project question with scenarios.
9. Fragment handling with lifecycle and communication between them.
10. Difference between live data and flow.
11. Difference between lazy and Lateinit.
12. Scope functions.
13. Rxjava related questions.
14. Data structure questions with its use cases like when we use LinkedList, ArrayList, hashmap, etc..
15. Dependency injection.
16. Caching the data not to repeat the API calls for similar data.
17. Kotlin coroutines.
18. Handler looper Thread related questions.
19. Unit test
20. Two programming problems.
21. Related to the time intervals given you need to find out the total spend time of the user on the particular video example -(0,10)(10,20)(15,25) etc…
22. Get the number that has a higher frequency in the given array and replace it with the given key or number.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Reliance\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Android: MVVM in Detail coroutines in detail Launch Mode Fragment lifecycle constraint layout Work Manager Marshmallow permission

Kotlin: mutable list and list Of def. Scope Function and difference. Java: HashMap and hash set Queue LinkedList and type

HashMap and Hash set

How to implement SDK?

Linked List, Recycler View, two fragment communication, handlers, coroutines, Val and var, latent and lazy, inline functions, unit Test cases

All the above topics are in detail should be explained

Android, Java and Kotlin related questions

kotlin, intent, MVVM, and one logic program

Question asked in interview: Reliance

Summary: Whole interview gone on the use cases and scenarios based questions

related to the components.

● Introduction and then questions start from the projects you have done.

● RecyclerView handling with different layout managers.

● Difference in between listview and recyclerview.

● Recyclerview internal working.

● Design patterns.

● MVVM related questions.

● Jetpack component with detailed explanation.

● How to handle the socket and how to deal with audio video calls because I

have done a project on that related to that particular project question with

scenarios.

● Fragment handling with lifecycle and communication between them.

● Difference between livedata and flow.

● Difference between lazy and lateinit.

● Scope functions.

● Rxjava related questions.

● Data structure questions with its use cases like when we use

linkedlist,arraylist,hashmap etc..

● Dependency injection.

● Caching the data not to repeat the api calls for the similar data.

● Kotlin coroutines.

● Handler looper Thread related questions.

● Unit test

● Two programming problems.

1. Related to the time intervals given you need to find out the total

spend time of the user on the particular video example

-(0,10)(10,20)(15,25) etc...

2. Get the number which has higher frequency in the given array and

replace it with the given key or number.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Thoughtwork\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Project-related questions suggest some alternate way of doing the same which u have done earlier in those projects.

Mvvm related questions(describe each component you use to achieve this).

Dagger-related questions(Annotations).

Multi-module architecture.

Make your own architecture without using any jetpack component which manages the lifecycle.

Webrtc related questions how we can share the screen how we give access overcall.

Threading how you achieve that in kotlin or in general if you have too many threads running how will you manage them

Question related to coroutines.

Synchronized keyword

How do you prevent from being another instance of the same class?

Question-related to the problem statement that you have 7 boxes in those boxes one is heavier. You need to find out the heavier one and tell me the number of iteration you need to achieve this.

Questions related to the parallel execution of threads.

Navigation components of the jetpack.

Paging

Database migration-related question

All question based scenario, like this is feature and you have to implement them this way.

So I want to do this in another way. (something like that).

Dagger

Stack and queue (scenario based and algo).

graphQL optimisation and rest api differences. (scenario based question).

Code optimisation

Memory optimisation

Local database (scenario based and you have to deepest knowledge)..

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_E Infochip\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Kotlin->

Coroutine question

MVVM

Livedata

Difference between set value and post value

Difference between listview and recyclerview

What is data class

What is lambda function

What is in kotlin which is not in Java

Service

Broadcast reciver

Fragment

Asked one scenario based question

Oops concept

Data binding

MVVM

Live Data

Android jetpack components

Java - >

Garbage collection related question

Oops related question->

Encapsulation

Polymorphism

Interface

Runtime polymorphism

Compile time Polymorphism

What is thread

Difference between thread and Runnable

How to secure your database

Proguard rules

Agile methodlogy

Difference between final, finally and finalize in java

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_OPPO\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

how do you efficiently check if there's an internet connectivity with or without using library from SDK itself.

How do you upload the image/video to the server if internet breaks in between or some unwanted error arises?

In the multiple bluetooth connected device how do you position the particular bluetooth device using bluetooth only?

How do you upload heave video file(ex 500MB) to the server?

heavy\*

Asking recently project - explanation

how multipart works -> scenario based question like - I have 1 gb video I want to upload to server, suppose I have upload 100mb data and after network lost due to bad connection, how to upload next 900 mb data. How to retrieve that. How to manage them.

services

activity with dialog life cycle.

retrofit call back and in which thread is working.

1.Previous company project explained.

2. Coroutines and implementation.

3.Lazy and latinit difference.

4. synchronous and asynchronous difference.

5.how to view model update.

6.Live data related question

**Xebia**

**sample questions for xebia**

1.write singleton class

2.write mvvm code

3.handle screen orientation cases

4.launch modes

5.diff b/w val and var

6.diff b/w val and const

7.what is lateinit

8.what is null safty

9.about dagger

10.what are coroutines

11.handle exceptions in coroutins

12.parrallel excution in coroutins

13.singleton in kotlin

14.sealed classes

15.diff b/w comapare and comparator

16.what hashcode method will do

L2:

1.tell about project

2.did u handled ui using html or native

3.how u handled sync something to server

4.particulaar time sync how u handled

5.different environments how u did in project

6.activity B activity A lifecycle flow of B and A activities

7.used dagger?

8.used Rxjava?

What is abstraction?

What is encapsulation?

What is runtime polymorphism?

Encryption methods in java

What are the difference between abstract class and interface?

What is loose coupling?

Explain MVVM architecture

What is live data?

What is singleton class?

How to save data while screen orientation occur?

What is hash map?

Difference between hashmap & hashset

What is coroutines?

Explain how to implement coroutines

What is dependancy injection?

-What is a benefit of using coroutines over threads

-String Interpolation in Kotlin

-extension function kotlin

-lambda expression

-http vs https difference

-visibility modifiers in Kotlin?

-What's the difference between "minifyEnabled" and “useProguard"