# Interview Questions

Easy

Medium

Hard

## Conversational:

### Kotlin

Question: What is Kotlin and why was it made

Answer:

* JVM based fully interoperable with Java
* Was made to increase java development speed by providing shorter syntax and additional features

Follow up:

Question: What are some Kotlin features that make development easier or faster compared to Java

Answer:

* Kotlin coroutines make non blocking asynchronous programming easier to write compared to reactive java which is written through call backs
* Kotlin provides null safety reducing errors and removing the need to work with Optionals
* Smart casts -> Allows casting of objects without explicity casting through type checking
* Reduces boiler plate and allows to write code in functional style
* Data classes with auto generated toString, equals, hashcode and copy methods

Question: What are Kotlin coroutines? (What advantages do they provide)

* Coroutines are an abstraction on threads allowing for asynchronous functions to be suspended and for the executing thread to execute some other function while waiting for the first to return. Same result can be achieved in Java using reactive programming but requires much more code and is written through callbacks instead of a synchronous way of programming.

Question: What are some changes that Kotlin introduced to classes compared to Java?

Answer:

* Primary Constructor: Secondary constructors need to delegate to primary, primary constructor can define instance variables
  + Q: What is the order of execution of constructors and init blocks?

Primary Constructor -> Init blocks and instance variables in order -> Secondary Constructor

* No more static variables, now have companion Object
  + Q: What if I want to still have a method defined as static? (why would you do this)

You can use @JvmStatic -> to be able to call the code in Java without referencing the companion, this will add the method to the companion and class as static.

* Data Classes -> mentioned above
* Sealed Classes -> prevent inheritance except for nested classes

Question: How do you define custom getters and setters in Kotlin?

Answer: Write get() and set(value) under variable -> can use field variable inside these methods to access the field itself

* Q: What is a backing field and when is it generated

Backing field is not generated if Setter and getter never reference it, if no setter present then not generated if getter does not reference it

Question: What is delegation in Kotlin

Answer:

* Delegation Allows to pass on responsibilities of a class to another class
* Dependency injection, when class implements another class and takes a dependency part of primary constructor, can be delegated to

Follow up What are some built in kotlin delegations?

* Lazy
* Vetoable
* Observable
* Map
* Not null

### Java

Question: What is an abstract class in java

A class that cannot be directly instantiated but can be extended

Abstract class can have abstract methods without a body

Follow up: What is the difference between abstact class and interface

* Interface only has abstract methods
* Class can implement many interfaces but only extend one class
* Interface can only have static variables

Question: How does a hashset know it already contains an object?

Answer:

* It uses the hashcode method of the object, the return is the position in the underlying array of the hashset that the set checks, it then uses .equals to handle collisions, if overriding .equals, make sure to override .hashcode

Question: When overriding .equals, why should you override .hashCode

Answer

* Hashcode is used by hash based collections like hashset, hashmap etc, two objects that return equals true should also return the same hashcode as that is what is used to check equality

Ask about abstract

### Java/TypeScript

Question: What is the purpose of npm run build step

* Browser can’t understand the js files you wrote (it does not understand modules) -> Actually now it does but before, npm run build will take your js files and combine them into one file which the browser can understand
* Compile JSX or other framework code into HTML JS and CSS

Question: What is typescript and how does it differ from javascript?

* Superset of JS that adds static typing
* Good for developers as it catches errors at compuletime

Question: What is a union in typescript

* Allows a variable to have more than one Type

Follow Question: In Typescript what is a discriminated Union

Answer: it is a type whose fields change depending on some discriminator, it is defined as multiple types joined using the | operator with each type sharing one field with a diffirent value, this field acts as the discriminator

Follow up: And how do you narrow down the type in your code?

You can check for the value of the discriminator in an if statememnt and the type will be narrowed

Question: "In TypeScript, can you explain how to create a function that both returns a Boolean value and infers the type of the object passed to it? Additionally, what is the term used to describe such a function?"

* Typeguard
* You write a function that returns a boolean but the return type is “passedInValue is Type”

Question: Explain the difference between an interface and type in typescript

* Interfaces can only represent objects while types can represent interfaces
* Interfaces can be modified later, types cannot

Question: What are generics in Typescript and how are they used?

* Generics creating components that work with any Type

Question: What is a conditional Type in typescript?

* Conditional types allow you to make types that depend on another type
* T extend string ? “string” : “number”

Question: What are decorators in TS, how are they used, and how to enable them?

* Decorators are functions prefixed with the @ symbol
* They allow you to modify classes, methods or properties during compile time
* They allow you to add metadata to classes and their properties and can be useful for logging, dependency injection, and adding repetitive functionality
  + Class decorators
  + Method decorators
  + Property Decorators
  + Accessor decorators
  + Parameter decorators
* Decorators need to be enabled in TS config by setting experimental decorators to true (or can pass flag to compiler)

### React

Question: What are react hooks?

Answer: React functions that allow functional component to use state and other features

Examples: useState, useEffect, useContext -> Anything else with use…

Question: What are controlled and uncontrolled components in react

* Controlled: Data is handled by react State
* Uncontrolled: Data is handled by the dom such as through refs

What are some advantages/disadvantages of uncontrolled?

Can improve performance

Easier integration with some libraries

Often used in Forms with default values

Disadvantages:

Goes against reacts philosophy and limited testing

Question: What is JSX/TSX?

JSX (javascript xml) is a syntactic enhancement to JS that gets compiled into javascript react objects (or other framework when using) it provides advantage of writing html looking code in js files

Question: What are props in react?

* Properties allow passing data from parent to child
* They are immutable

Follow: How to pass data from chuld to parent?

* + Callback functions

### Angular

Question: What are different kinds of directives that exist in angular

* Component Directives – Used to create reusable ui components
* Attribute Directives – used to add attributes to an element
* Structural Directives – used to modify structure of dom -> ngFor

Question: What is a service in angular and what is it used for

* Singleton instance
* Dependency Injecton

Question: What is Angular Routing and how does it work?

Angular Routing is a mechanism for navigating between different components while maintaining the state of the application. It allows developers to define navigation paths and associate them with specific components. Angular's Router listens for browser URL changes and maps them to the corresponding component to be displayed.