Weekend 5 Homework?

1.What is the difference in late initialization and lazy initialization?
Sub.) what is late initialization
Late Init is what we use when we want our variable to be initialized later and we as the developer will be sure the variable will not be null when accessing it.
Used by the lateinit Keyword and a var

Sub.) what is lazy initialization
lazy --val,, lateinit--var*
First of all, **Lazy** can only be used for **Val**, Whereas lateinit can only be applied to **Var.**
. Variables that are initialized by lazy will not have memory allocated until the variable is used.
Lazy() is a function that takes a lamba, returns an instance if lazy

2. What are the advantages of Kotlin over Java?
***NPE,, boiler plate code,, lambda function,,extension function**
Kotlin has a few advantages over java, some examples are -
Kotlin Tried to eliminate Null pointer Exception
Kotlin utilizes Dynamic typing

Kotlin HasReduced Boiler Plate Code
-Kotlin utilizes Lamda Heavily
-Has no Primitives
-has inline functions
-has Extension functions

3. What are checked and uncheck exceptions? Which is not supported by Kotlin?

Sub.) What is an Exception?

An exception is an unwanted or unexpected event that occurs while a program is executed.
(Handled in Java by try, catch, throw,throws and finally)

Sub.) What is a checked Exception?

Checked Exceptions are exceptions which are checked at compile time. Example Filenotfound, NoSuchMethod exception

Sub.)What is an Unchecked Exception?

Unchecked exceptions are exceptions that are not checked at compile time. It is up to us to specify and catch the exception.

**Example is a number/0 throws an arithmetic exception , nullPointerException
`Example >throw new IllegalArgumentException("Invalid age"); **`

Sub.) What exception is not Supported by Kotlin?
Kotlin does not have Checked exceptions

4. What is Model View Presenter?

MVP is an architectural design pattern that is used for maintainability and to improve separation of concerns

Sub.) What is the role of Model?

****Data layer,,handle business logic,, communication with database and network layers****

Model is the data layer. Responsible for handling business logic and communication with database and network layers.

Sub.) What is the role of View?

**** UI layer,,display data,, notifies presenter of user actions**

View is the UI layer, Dispays data and notifies the presenter about user actions

Sub.) What is the role of Presenter?

****retrieves data from model,, applies UI logic and manages the states of views**

Retrieves data from the model, applies ui logic and manages the state of the views. The presenter decides what to display and reacts to user input notifications from the view

Sub.) What Class must be implemented/ defined for a view/Presenter relationship?

A contract interface class must be defined to make the connection easier to understand

1to1, Every View has a presenter

5. What is Model View ViewModel?

****A view can have many models(Not 1 to 1),, Use Live Data****

Sub.) What is a view in MVVM?

View is the collection of UI elements elements which also receive user input. Responsible for defining the structure, layout and appearance of what the user sees

Sub.) What is a Model in MVVM?

Model classes are non -visual classes that encapsulate the apps data (used in conjunction with services and repositories

Sub.) What is a ViewModel in MVVM?

View Model implements properties and command to which the view can bind to. Also notifies the view of any state change through change notification events

Review

What is the difference between a constant val and a val?

Const vals are constants/variables that must be assigned at compile time(right away)

Vals are variables that can be assigned at runtime.

What do i use lazy initialization for?

Kotlin equivalent as java static?

Companion Object

How do you refactor java code to Kotlin code?

Copy and paste and Kotlin will ask to convert to Kotlin
Right click and select convert java file to kotlin file

What is an extension function?
**** way of extending existing class, without using inheritance**
Adding a method on the fly to any class.
They are a handy way of extending existing class with a new functionality without using inheritance of any forms of decorator pattern

How do you add a fragment to an activity?

Architecture Design

-we use it for separation of concern and maintainability

Most common architecture-

MVC- model view controller(oldest

MVP Model

Presenter- logic, anything not directly view,model,control
*Contract - interfaces
View - User sees
Model - pojo, dataclasses, datasources

- One to one relationship between view and presenter(every view has a presenter and viceversa)

MVVM(Model View ViewModel

-For communication MVVM uses DataBinding(Arch.components)
-View and viewmodel has a one to many relationship(a view can have many viewmodel
- abstract code to xml
- can use livedata