**Assignment Module 4 – Android Studio and Android App Structure**

**1)Explain the Android app structure in detail, including the purpose of the AndroidManifest.xml file, Gradle, and various directories (e.g., res, src).**

**The **AndroidManifest.xm**l**

🡪The **AndroidManifest.xml** file is a very important part of every Android app. It tells the Android operating system key details about the app

🡪android manifest xml file is requrid in all app

**🡪App Name and Icon**: What the app is called and what its icon looks like

🡪Intent-filter: intent filter use to lunch activity first

**Gradle**

Gradle is a tool that helps developers automatically build, test, and deploy their software, making the process faster and easier.

**Directories:**

**src/**:

* Contains all your Kotlin source code files.
* **Kotlin files** are stored with the .kt

**res/ (Resources)**:

drawable/ : drawable store image ,background image,and icons

Layout/: Contains the XML layout files that define the UI structure of your app.

Values/ :it is used to store string,colours dimensions,styles

Mipmap/ : use to store app icon

1. **Write an essay on the different layout types in Android (Linear Layout, Relative Layout, Constraint Layout). Compare their usage and performance.**

**Linear Layout**

A Linear Layout is a view group in Android that arranges its child views in a single direction, either vertically or horizontally.

Use android:orientation="vertical" for vertical stacking.

Use android:orientation="horizontal" for horizontal alignment.

Child views can be given weights (layout\_weight) to proportionally distribute space.

It’s simple to use but can become inefficient for complex layouts due to nested views.

**Relative Layout:**

A Relative Layout is a type of layout commonly used in user interface design, especially in frameworks like Android development.

It allows you to position and size elements relative to one another or relative to the parent container.

The key feature is that elements' placement is determined by their relationship to other elements or the parent layout, not by absolute positions.

**ConstraintLayout**

ConstraintLayout is like a super-flexible canvas for designing app layouts in Android.

Instead of stacking things in a line (like LinearLayout) or positioning them relative to other views (like RelativeLayout),

it gives you total freedom to arrange your buttons, text, images, and everything else exactly where you want them.

**3)Explain the differences between Fragment and Activity. How does Android handle fragment lifecycle differently from activity lifecycle?**

**What’s the difference between a Fragment and an Activity?**

**Activity:**Think of it as a full-screen page in your app. It’s the main thing users interact with like the homepage, settings screen, or profile page. Activities are independent and can exist on their own.

**Fragment**:

Fragments are smaller, reusable pieces of UI that live inside an Activity.

They’re like sections or widgets of a screen. For example, in a shopping app, an Activity could show the product details, and the Fragment might handle just the product image gallery or the reviews section

Here’s the main difference:

* Activities are like the whole house.
* Fragments are like the individual rooms inside the house

**How is the lifecycle of a Fragment different from an Activity**

#### ****Activity Lifecycle:****

1. onCreate(): This is where the Activity sets up its UI.
2. onStart() **/** onResume(): The Activity becomes visible and interactive.
3. onPause() **/** onStop(): The Activity is no longer in focus or visible.
4. onDestroy(): The Activity is shut down.

#### ****Fragment Lifecycle:****

1. onAttach(): The Fragment gets connected to its Activity.

2. onCreateView(): The Fragment sets up its own little part of the screen.

3. onDestroyView(): The UI part of the Fragment is cleaned up (but the Fragment still exists in memory).

4. onDetach(): The Fragment is completely removed from the Activity

#### ****The Key Difference:****

* **Activities** manage the entire app screen and its lifecycle independently.
* **Fragments** rely on the Activity for their lifecycle. If the Activity is paused or destroyed, all its Fragments are paused or destroyed too.

Activities are like whole pages in an app, while Fragments are like sections or components of those pages. Imagine an Activity as the full page of a book, and each Fragment as a smaller part of the page, like a paragraph or a picture, that can be used in different parts of the app.