Md Rahman

Week1Day4 Research

1. Define the following terms: View, ViewGroup, View Hierarchy.

1. The UI in an app is developed with a collection of Views and Viewgroup objects. All the activities (screens) inside an app hold various UI components such as EditText, ViewText, Button, etc. and these components are sub-classes of view. Essentially, views are all the types of elements a user can interact with on a single activity screen.
2. ViewGroup is a special view that can contain other views, and is the base class for layouts and various containers such as Linear Layout, Relative Layout, and Grid View.
3. A view inside another View creates what’s known as a View Hierarchy, where the most outer view becomes the parent and the inner views are the children.

2. Explain in detail how the following layouts render, a what unique items each has that must be implemented: Constraint, Linear, Coordinator, Grid and Relative?

1. Constraning Layout has only one rendering process because there are no nested ViewGroup, making it very fast. Based on the specifications provided by the developer such as toStartOf, toEndOf, toTopOf, etc. in relation to the parent view.
2. Linear Layout aligns child Views and viewgroups in a horizontal or vertical direction specified using android:orientation. Children in a linear layout are stacked one after the other, meaning a vertical list of elements will only have one child per row. Certain attributes can be assigned to each children such as android:layout\_width and android:layout\_height.
3. Coordinator Layout - Perhaps the most special layout because it can allows for very complex customizations. The developer may create their own custom constraint behaviors and may tune the viewgroup(s) to however they see fit using child anchors.
4. Grid Layout - Gives you ability to create grid based layouts using a single root view. Used to place all child elements into a set of horizontal and vertical lines. We can use android:columnCount and android:rowCount
5. Relative Layout - Specify how child views are positioned relative to each other. For example, we can align two elements side by side by using android:layout\_align, one above the other using android:Layout\_above, or android:layout\_centerInParent to center child horizontally and vertically within its parent.

3. What are Listeners?

Listeners are an interface in the View class which contain certain call-back methods which are triggered when the user interacts with the registered view. For example, a button may have an onClickListener() attached which is triggered when the user clicks it. The developer can define what actions happen when it’s clicked. This happens using three particular steps:

1. Event Listener - The interface itself that contains the call-back method.
2. Event Listener Registration - The link that connects the event handler (call-back method) to the Event Listener, so it will get called when the user does something
3. Event Handler - The particular call-back method itself, such as onClick(), onDragListener(), etc.

4. How does Java garbage collection work?

1. Garbage collection is the process of memory management, and in Java it’s done automatically. When a Java program is compiled to bytecode and run on the JVM, objects are created on a portion of memory specifically dedicated for that program, known as the heap. However, as the program changes or updates over time, certain objects will no longer be used, so instead of taking up unnecessary space, the garbage collector that’s built into the JVM finds those unused objects and deletes them.

5. Explain the software development lifecycle.

The framework defining that tasks performed in each step when developing software. This creates a structured approach for creating and maintaining software and is the first thing new software developers are taught. While there are different SDLC models, there are six important stages that each SDLC model will adopt:

1. Planning and Requirement analysis - Getting input from customers, sales and other experts in your industry to determine if the project is feasible from an economic and technical point of view.
2. Defining Requirements - Getting input from your client(s) to determine exactly what features they want to figure out the feasibility of implementing said features.
3. Project Architecture - After approving the requirements, an architecture must be defined and followed for developing the project with these constraints (time, budget, features) in mind.
4. Developing the Product - Where the actual development happens by the team. Everyone must agree upon the development tools such as what language(s) to use.
5. Testing the Product - Testing the product. Not only are tests done by the developers but others such as QA.
6. Deployment - After passing all tests, the product is formally released to the client.