

1. Define the following terms: View, ViewGroup, View Hierarchy.
  - a. The View class is the building block for user interface components. A View is a rectangular area on the screen. Within this area is where drawing and event handling takes place. The View class is the base class of widgets. Widgets are used to create the interactive UI components.
  - b. ViewGroup is an invisible container that holds the Views and other ViewGroups. ViewGroups are the parent to views(children).
  - c. View Hierarchy is the structure of nested Views, ViewGroups, and Layouts. Views can be nested in ViewGroups, ViewGroups can be nested in Layouts, and Layouts can be nested in other Layouts. The visual representation of the nesting creates View Hierarchy Tree.
2. Explain in detail how the following layouts render, what unique items each has that must be implemented: Constraint, Linear, Coordinator, Grid and Relative?
  - a. Constraint: This layout render views in relation to other children or the parent similar to relative layout. Constraints allows for the building of large complex UI with a flat hierarchy, no nesting. Constraint layout is used by the Layout Editor. A UI can be built entirely from the Layout Editor. Each view in the Constraint layout must have at least one horizontal and one vertical constraint added. Each constraint is connected or aligned with the parent, another child or some other type of guidelines. As such, each view must have a constraint for each axis in the layout.
  - b. Linear : This layout render views in either a single column or row. The orientation of the layout defines which way and must be specified. Linear layouts can be nested to establish multiple children on one row or column. A weight can set to determine the amount of space between each child.
  - c. Coordinator: Coordinator layout is container for views that interact with each other. This layout is generally used as the root for all views or nested into other layouts and contain the views that need to interact. This layout contains the attributes to allow the views within it the behaviors to interact with other children and the parent.
  - d. Grid: This layout renders views based on a rectangular grid. The grid contains a number of horizontal and vertical lines grid indices. The indices separate the viewing area into boxes called cells. Children are placed in cells and can be place in cell by developer or automatically if not set.
  - e. Relative: This layout render views based on the position of other children or relative to the parent. By default, views are rendered at the top left corner of the layout. Each child must have its attribute IDs set properly to ensure accurate render location. Due to default, all but one view must have a relative attribute to eliminate views from stacking onto each other.
3. What are Listeners?

- a. Listeners are interfaces that captures user interactions with the UI. When a user touches a button, the interaction could be caught by the button methods, but then each button would have to extend and override these methods in order to capture the interaction. In place of all this coding, a listener can be used. The listener can then capture the input and relay the proper instruction of the interaction.
- 4. How does Java garbage collection work?
  - a. In programming languages like C and C++, the programmer is required to create new objects and destroy unused objects. Programmers often forget to and may encounter an OutOfMemory error due to unused objects not being destroyed. With the Java Garbage Collection (GC), user is no longer required to delete unused objects. The GC is a part of the Java Virtual Machine and will destroy objects that are unreachable. An object is unreachable if it is nullified, the reference variable is reassigned, an object created in a method, on an Island of Isolation. By destroying the object dynamically, memory is freed and programs are less likely to run out of memory.
- 5. Explain the software development lifecycle (SDLC).
  - a. The SDLC is a procedure of steps for developing software. There are many different SDLC put in practice, but the steps may be done in different orders or multiple steps done concurrently. In general they all incorporate the following steps
    - i. Planning: This is the first step. This is where the problem domain is established and research and requirement gathering takes place. A scope of the project is created and documented.
    - ii. Implementation: After planning for the project, the developers begin coding.
    - iii. Testing: After a reasonable amount or all of the expected coding is done, developers begin testing and debugging the written code.
    - iv. Documentation: This step involves documenting what steps were taken, what could be improved on the software or during development, drafting documents for the software, and possibly coding an API.
    - v. Deployment: The software had been approved and is released for use.
    - vi. Maintenance: This when the software is review after being deployed. The software may be updated for better performance or new requirements via patches. The software may not have continued support or may start the cycle all over again for the next version.