

Android Assignment

Research

- a. What is Android? Who created it? What are Android Apps?

Android is a mobile operating system based on a modified version of the Linux kernel and other open source software, designed primarily for touchscreen mobile devices such as smartphones and tablets. Andy Rubin, Nick Sears and Rich Miner created it. An Android app is a software application running on the Android platform.

- b. What is the software used in the development of Android Apps?

Android Studio is the software used in the development of Android Apps.

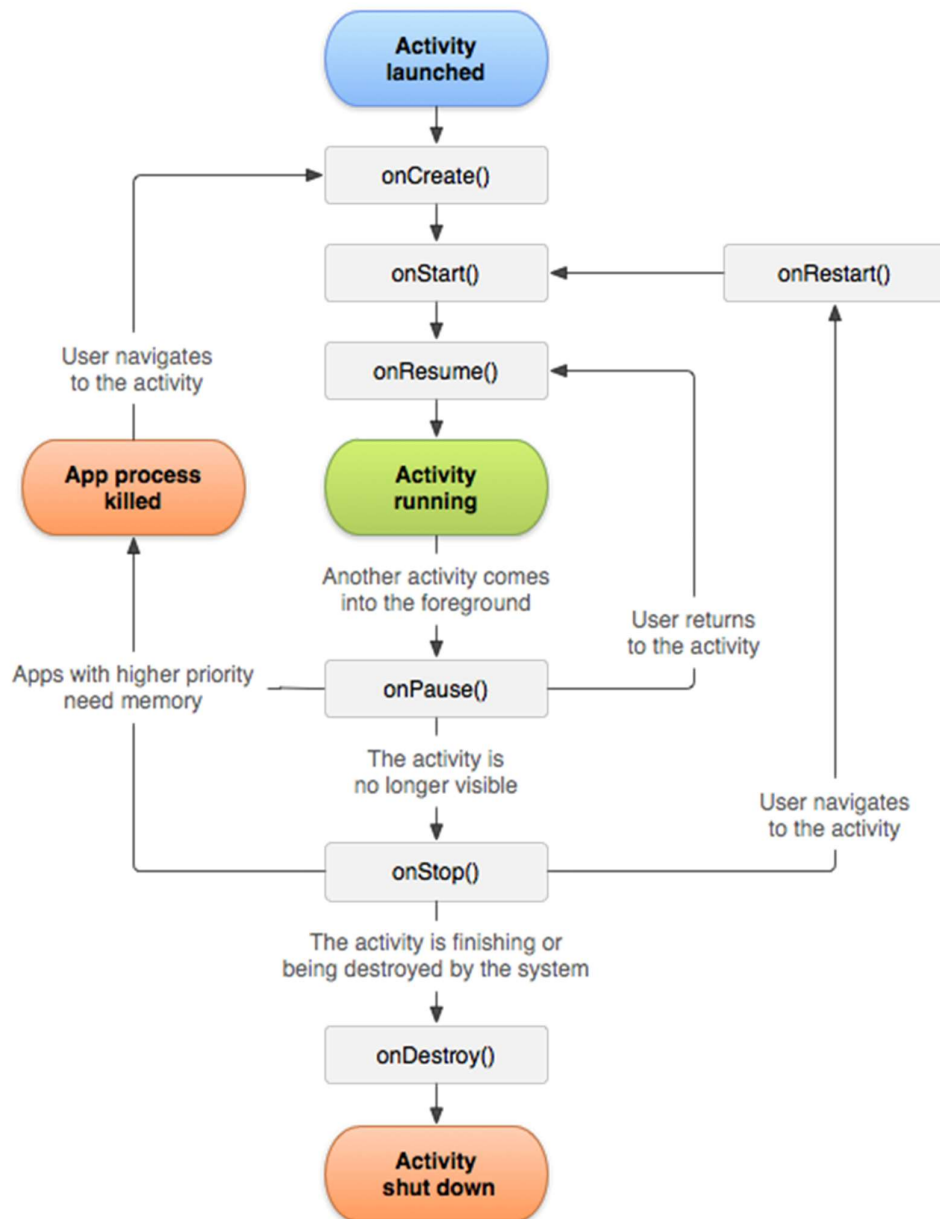
- c. Which are the languages commonly used in the development of android apps? Which language does InstiApp use?

Android apps are commonly written using Kotlin, Java, and C++ languages. InstiApp uses Java.

- d. What is the activity cycle of a basic Android application? Diagrams/flowcharts preferred.

As a user navigates through, out of, and back to your app, the Activity instances in your app transition through different states in their lifecycle. The Activity class provides a number of callbacks that allow the activity to know that a state has changed: that the system is creating, stopping, or resuming an activity, or destroying the process in which the activity resides. To navigate transitions between stages of the activity lifecycle, the Activity class provides a core set of seven callbacks: `onCreate()`, `onStart()`, `onResume()`, `onPause()`, `onStop()`, `onRestart()` and `onDestroy()`. The system invokes each of these callbacks as an activity enters a new state.

1. `onCreate()`: You must implement this callback, which fires when the system creates your activity. When `onCreate()` finishes, the next callback is always `onStart()`.
2. `onStart()`: As `onCreate()` exits, the activity enters the Started state, and the activity becomes visible to the user. This callback contains what amounts to the activity's final preparations for coming to the foreground and becoming interactive.
3. `onResume()`: The system invokes this callback just before the activity starts interacting with the user. The `onPause()` callback always follows `onResume()`.
4. `onPause()`: The system calls `onPause()` when the activity loses focus and enters a Paused state. Once `onPause()` finishes executing, the next callback is either `onStop()` or `onResume()`, depending on what happens after the activity enters the Paused state.
5. `onStop()`: The system calls `onStop()` when the activity is no longer visible to the user. The next callback that the system calls is either `onRestart()`, if the activity is coming back to interact with the user, or by `onDestroy()` if this activity is completely terminating.
6. `onRestart()`: The system invokes this callback when an activity in the Stopped state is about to restart. This callback is always followed by `onStart()`.
7. `onDestroy()`: The system invokes this callback before an activity is destroyed. `onDestroy()` is usually implemented to ensure that all of an activity's resources are released when the activity, or the process containing it, is destroyed.



e. What are 5 different UI elements in an android app? One example is a “TextView”.

1. *TextView*: This control is used to display text to the user.
2. *Button*: A push-button that can be pressed, or clicked, by the user to perform an action.
3. *CheckBox* : An on/off switch that can be toggled by the user. You should use check box when presenting users with a group of selectable options that are not mutually exclusive.
4. *ProgressBar*: The ProgressBar view provides visual feedback about some ongoing tasks, such as when you are performing a task in the background.
5. *Spinner*: A drop-down list that allows users to select one value from a set.

- f. [BONUS]What are some of the salient features of those languages (part c)? How similar are they to C++?

Java:

1. Java is secure (no threat to security because nothing gets executed outside the JVM)
2. Java supports Object Oriented Paradigm.
3. Java's core features are complete and vast. Also, they're regularly updated and maintained by oracle.
4. Java is platform independent.
5. Outside the core library, java has many frameworks and classes for features like networking, threading, IO operations and thus, programmers can leverage these qualities in their apps.
6. Java is open source.

Kotlin: Kotlin is increasingly replacing Java because -

1. Kotlin allows writing less code
2. It solves developer challenges
3. Adopting Kotlin is easy
4. Kotlin is fully compatible with Java
5. It imposes no runtime overhead
6. Kotlin has a strong community
7. Kotlin suits for the multi-platform development
8. Kotlin development offers more safety

Comparison with C++ :

Task

1. Read about relative and linear layouts and how they are used to design the UI of Apps.
2. Now suppose you want to design the landing screen / dashboard of Book-ed!, what do you think should be the various features of that screen?
3. Draw a schematic diagram of the screen. You can do this using a pen and paper or use online android prototyping tools.
4. Now break down your design into various layouts and elements. Clearly mark what is a linear layout, what are the various elements being used in your design. For example if there is a piece of text somewhere on the screen, that part would be the “TextView”.
5. [BONUS] Install “Android Studio” on your laptop with all the necessary requirements and run the starter app on your android device.
6. [BONUS] Write the code for the dashboard which you just designed/prototyped.