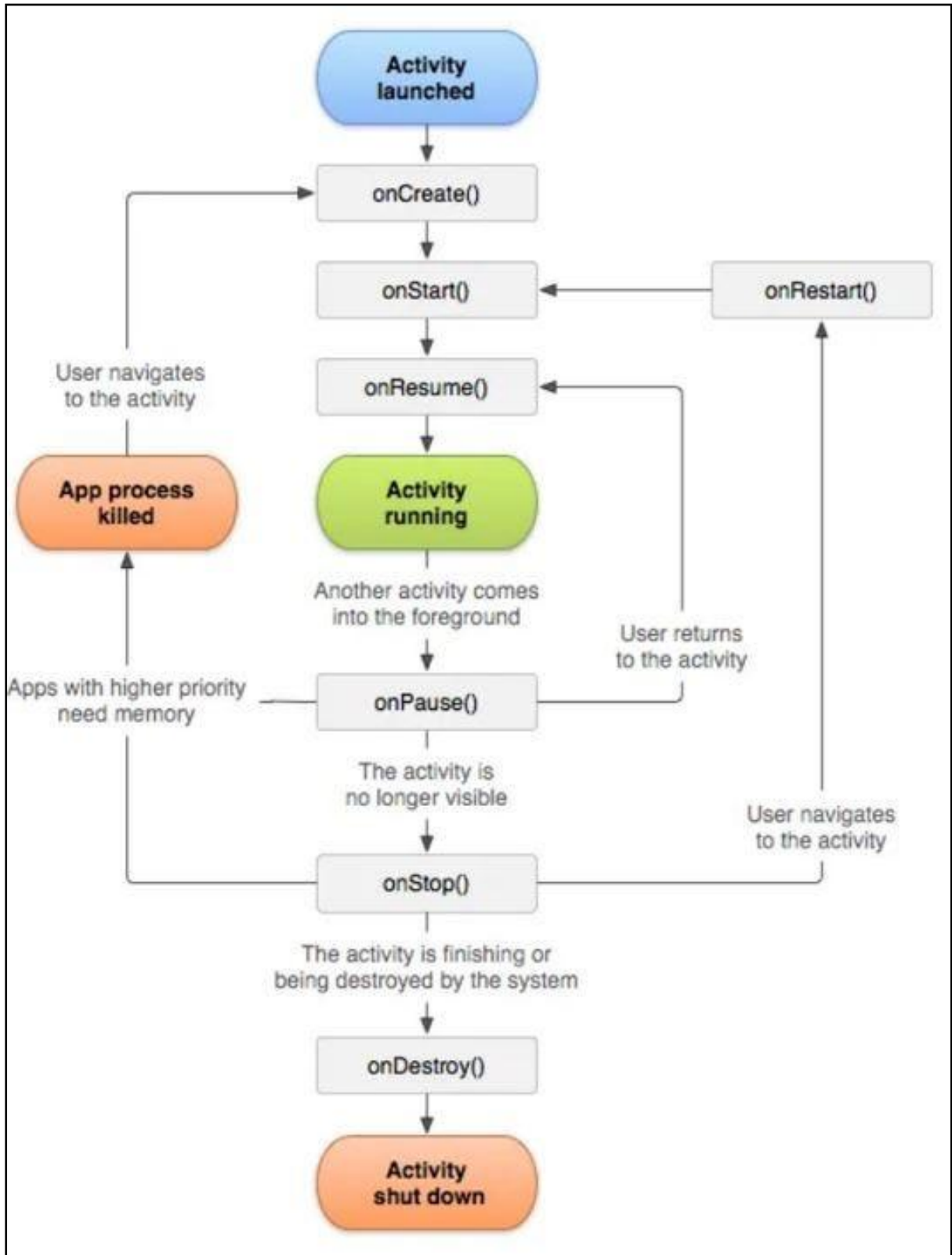




Definition

The **Activity Life Cycle** in **Android** refers to the **different states through which an Activity passes from the time it is created until it is destroyed**. Android manages these states using a set of **callback methods**, which allow developers to control what happens when an Activity starts, pauses, resumes, stops, or ends.

In simple words, the Activity Life Cycle defines **how an Activity behaves when a user opens the app, switches between apps, receives a phone call, or closes the app**.

States and Methods of Activity Life Cycle

1. onCreate()

- Called when the activity is created for the first time.
- Used to initialize components like UI layout, variables, databases, listeners, etc.
- Example use: setContentView() to load the XML layout.

2. onStart()

- Called when the activity becomes visible to the user.
- The activity is not yet interactive.
- Suitable for preparing the UI and starting simple tasks.

3. onResume()

- Called when the activity comes to the foreground and the user can interact with it.
- This is where the activity is actively running.
- Resume animations, sensors, or camera here.

4. onPause()

- Called when the activity is partially hidden (e.g., a dialog appears or another activity comes in front).

- Used to pause ongoing tasks, save small data, or stop animations.
- Should be quick because the next activity is waiting.

5. onStop()

- Called
- when the activity is no longer visible to the user.
- Used to release heavy resources like database connections, GPS, or background threads.

6. onRestart()

- Called after onStop() when the activity is restarting again.
- Followed by onStart() and onResume().

7. onDestroy()

- Called before the activity is destroyed.
- Final cleanup of resources.
- May not always be called if the system kills the process.