



# **SNS COLLEGE OF TECHNOLOGY**

**Coimbatore-35**  
**An Autonomous Institution**



Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A++' Grade  
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

## **DEPARTMENT OF INFORMATION TECHNOLOGY**

### **MOBILE APPLICATION DEVELOPMENT**

III YEAR - VI SEM

#### **UNIT 2 – BUILDING BLOCKS OF MOBILE APPS – I**

Topic 4 – Activity- states and life cycle



# Activity- states and life cycle



## Activity- states and life cycle

- Android system initiates its program with in an Activity starting with a call on onCreate() callback method.
- There is a sequence of callback methods that start up an activity and a sequence of callback methods that tear down an activity as shown in the below Activity life cycle diagram
- The Activity class defines the following call backs i.e. events.
- You don't need to implement all the callbacks methods.
- However, it's important that you understand each one and implement those that ensure your app behaves the way users expect.



# Activity- states and life cycle



Sr.No	Callback & Description
1	<code>onCreate()</code> This is the first callback and called when the activity is first created.
2	<code>onStart()</code> This callback is called when the activity becomes visible to the user.
3	<code>onResume()</code> This is called when the user starts interacting with the application.
4	<code>onPause()</code> The paused activity does not receive user input and cannot execute any code and called when the current activity is being paused and the previous activity is being resumed.
5	<code>onStop()</code> This callback is called when the activity is no longer visible.
6	<code>onDestroy()</code> This callback is called before the activity is destroyed by the system.
7	<code>onRestart()</code> This callback is called when the activity restarts after stopping it.



# Activity- states and life cycle



## Example

This example will take you through simple steps to show Android application activity life cycle.

Step	Description
1	You will use Android studio to create an Android application and name it as HelloWorld under a package com.example.helloworld as explained in the Hello World Example chapter.
2	Modify main activity file MainActivity.java as explained below. Keep rest of the files unchanged.
3	Run the application to launch Android emulator and verify the result of the changes done in the application.



# Activity- states and life cycle



An activity class loads all the UI component using the XML file available in res/layout folder of the project. Following statement loads UI components from res/layout/activity\_main.xml file:

```
setContentView(R.layout.activity_main);
```

An application can have one or more activities without any restrictions. Every activity you define for your application must be declared in your AndroidManifest.xml file and the main activity for your app must be declared in the manifest with an `<intent-filter>` that includes the MAIN action and LAUNCHER category as follows:



# Activity- states and life cycle



```
•<manifest xmlns:android="http://schemas.android.com/apk/res/android"
•  package="com.example.tutorialspoint7.myapplication">
•
•  <application
•    android:allowBackup="true"
•    android:icon="@mipmap/ic_launcher"
•    android:label="@string/app_name"
•    android:supportsRtl="true"
•    android:theme="@style/AppTheme">
•    <activity android:name=".MainActivity">
•      <intent-filter>
•        <action android:name="android.intent.action.MAIN" />
•
•        <category android:name="android.intent.category.LAUNCHER" />
•      </intent-filter>
•    </activity>
•  </application>
•
•</manifest>
```



# Activity- states and life cycle

