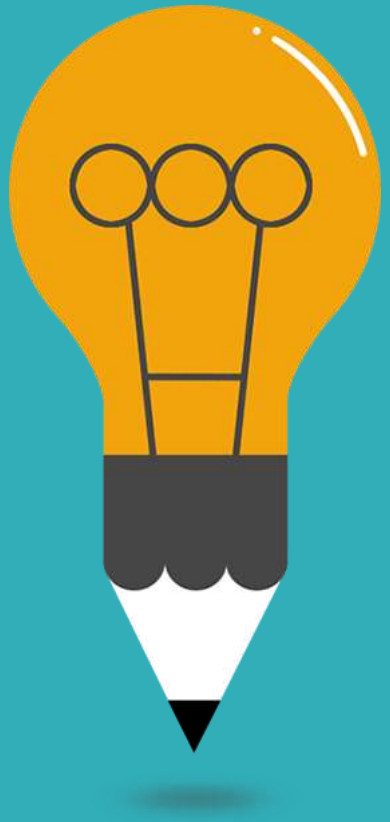
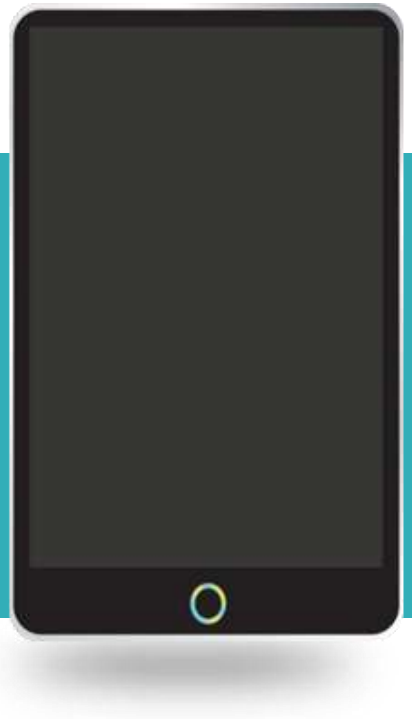


Android Activity Lifecycle

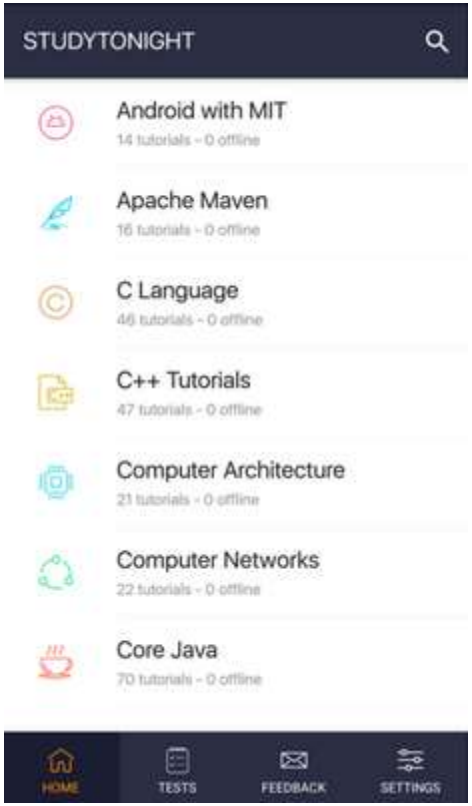
What is an Activity in Android?

- An **activity** is the single screen in Android.
- It is like window or frame of Java
- By the help of activity, you can place all your UI components or widgets in a single screen.



Activity in Android

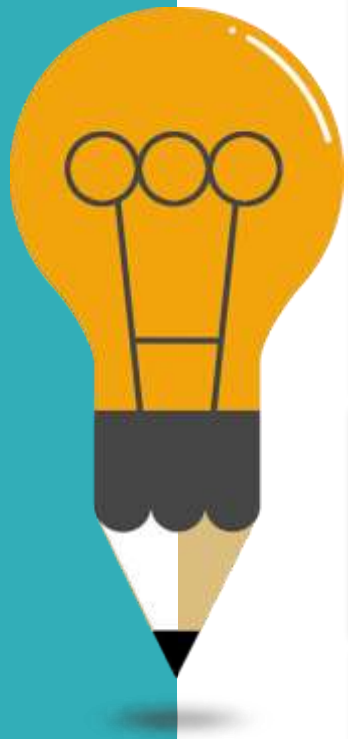
Screen



Java — **Activity**

XML — **Layout**

Different States



01 ACTIVE

When an Activity is in active state, it means it is active and running. It is visible to the user and the user is able to interact with it.

02 PAUSED

An activity being in this state means that the user can still see the Activity in the background such as behind a transparent window or a dialog box i.e it is partially visible. The user cannot interact with the Activity until he/she is done with the current view.

03 STOPPED

When a new Activity is started on top of the current one or when a user hits the Home key, the activity is brought to Stopped state.
The activity in this state is invisible, but it is not destroyed.

04 DESTROYED

When a user hits a Back key or Android Runtime decides to reclaim the memory al located to an Activity i.e in the paused or stopped state, It goes into the Destroyed state. The Activity is out of the memory and it is invisible to the user.

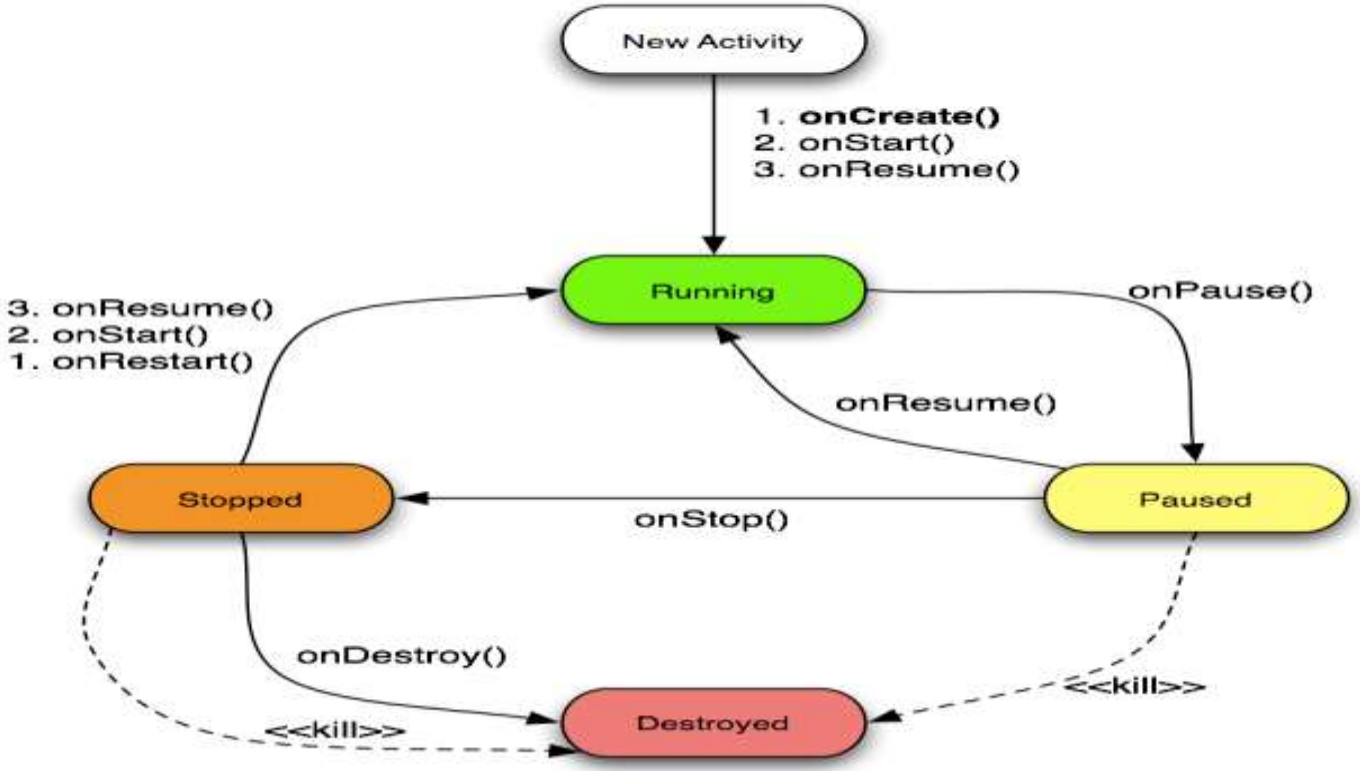
Activity Lifecycle methods

1. OnCreate( )
2. OnStart( )
3. OnResume( )
4. OnPause( )
5. OnStop( )
6. OnRestart( )
7. OnDestroy( )

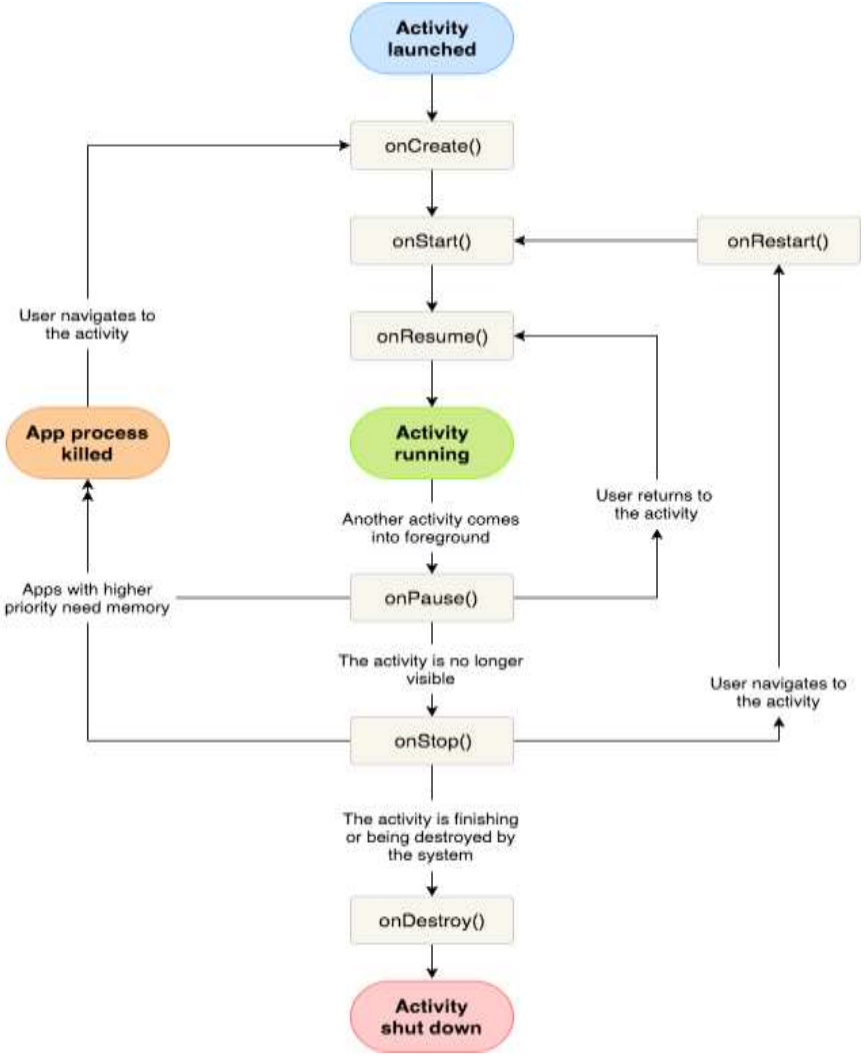
Activity Lifecycle methods

Method	Description
onCreate	called when activity is first created.
onStart	called when activity is becoming visible to the user.
onResume	called when activity will start interacting with the user.
onPause	called when activity is not visible to the user.
onStop	called when activity is no longer visible to the user.
onRestart	called after your activity is stopped, prior to start.
onDestroy	called before the activity is destroyed.

Activity Lifecycle



Lifecycle Diagram



activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/
res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="example.javatpoint.com.activitylifecycle.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```

MainActivity.java

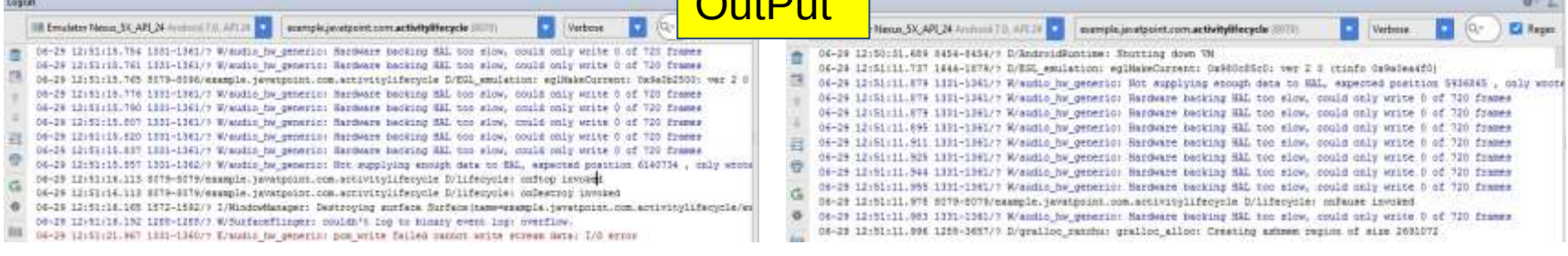
```
package example.javatpoint.com.activitylifecycle;

import android.app.Activity;
import android.os.Bundle;
import android.util.Log;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d("lifecycle","onCreate invoked");
    }
    @Override
    protected void onStart() {
        super.onStart();
        Log.d("lifecycle","onStart invoked");
    }
    @Override
    protected void onResume() {
        super.onResume();
        Log.d("lifecycle","onResume invoked");
    }
@Override
    protected void onPause() {
        super.onPause();
        Log.d("lifecycle","onPause invoked");
    }
    @Override
    protected void onStop() {
        super.onStop();
        Log.d("lifecycle","onStop invoked");
    }
    @Override
    protected void onRestart() {
        super.onRestart();
        Log.d("lifecycle","onRestart invoked");
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
        Log.d("lifecycle","onDestroy invoked");
    } }
```

OutPut



Thank you