SMART DEVICE PROGRAMMING

6133

ACTIVITY IN ANDROID

- ✓ An Android activity is one screen of the Android app's user interface.
- ✓ In other words, building block of the user interface is the activity.
- ✓ Activity class is a pre-defined class in Android
- ✓ An Android app may contain one or more activities, meaning one or more screens.
- ✓ The Android app starts by showing the main activity, and from there the app may make it possible to open additional activities .
- ✓ An activity provides the window in which the app draws its UI.

- ✓ To create an activity, you create a Java class that extends the Activity base class:

```
package net.learn2develop.Activities;
import android.app.Activity;
import android.os.Bundle;
public class MainActivity extends Activity
{
        /**Called when the activity is first created.*/
      public void onCreate(BundlesavedInstanceState)
          {
              super.onCreate(savedInstanceState);
              setContentView(R.layout.main);
          }
 }
```

- ✓ Your activity class would then load its UI component using the XML file defined in your res/layout folder.
- ✓ In this example, you would load the UI from the main.xml file:

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

- ✓ Every activity you have in your application must be declared in your AndroidManifest.xml file, as follows.

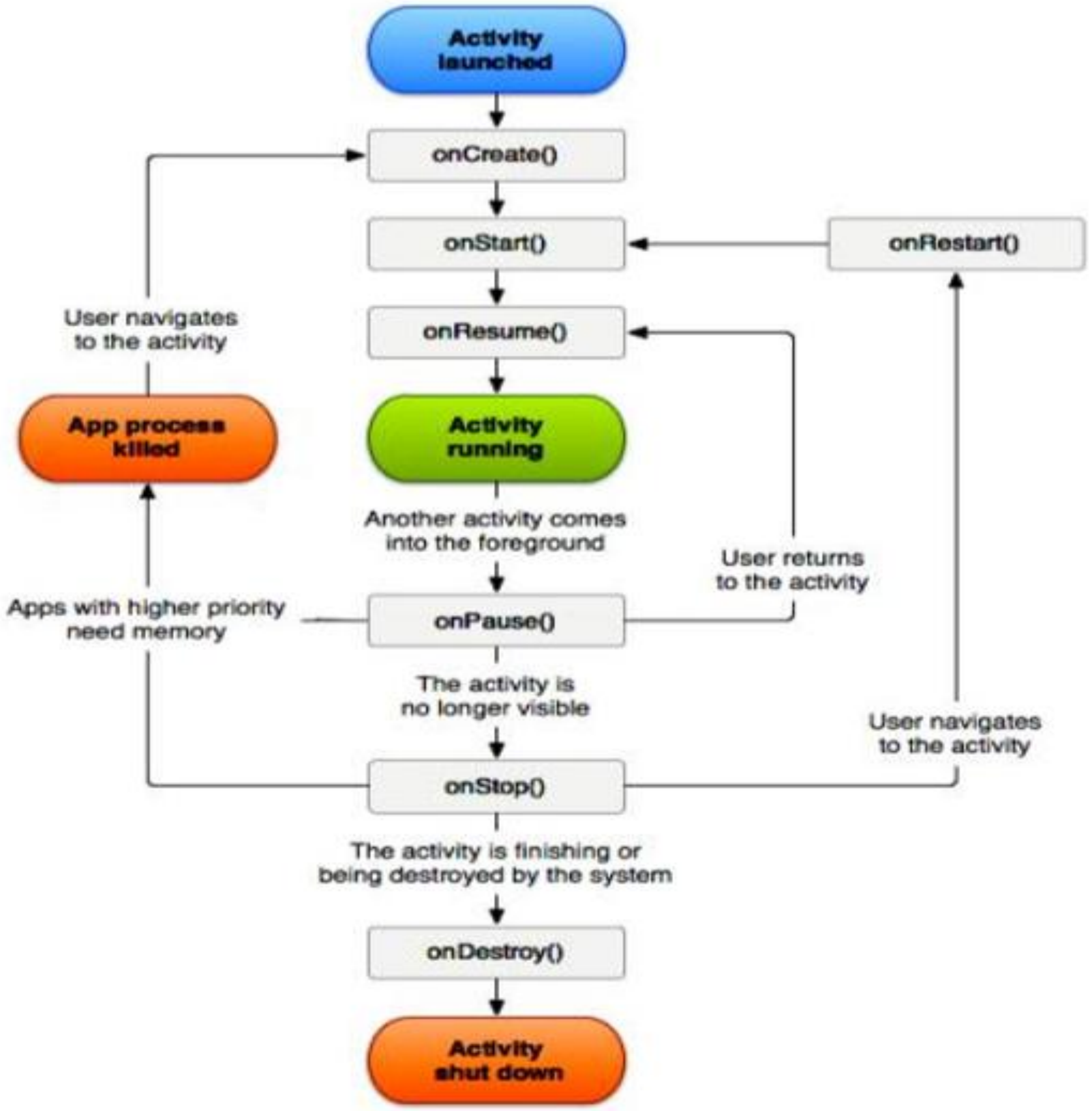
```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="net.learn2develop.Activities"
      android:versionCode="1"
      android:versionName="1.0">
    <application android:icon="@drawable/icon"
        android:label="@string/app_name">
        <activity android:name=".MainActivity"
                  android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category
                    android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
    <uses-sdk android:minSdkVersion="9" />
</manifest>
```

LIFE CYCLE OF AN ACTIVITY

- ✓ Android Activity Lifecycle is controlled by 7 methods of android.app.Activity class.
- ✓ The 7 lifecycle method of Activity describes how activity will behave at different states.
- ✓ The Activity base class defines a series of events that governs the life cycle of an activity.
- ✓ The activity created for you contains the onCreate() event. Within this event handler is the code that helps to display the UI elements of your screen.

✓ The Activity class defines the following events:

- onCreate() — Called when the activity is first created
- onStart() — Called when the activity becomes visible to the user
- onResume() — Called when the activity starts interacting with the user
- onPause() — Called when the current activity is being paused and the previous activity is being resumed
- onStop() — Called when the activity is no longer visible to the user
- onDestroy()— Called before the activity is destroyed by the system (either manually or by the system to conserve memory)
- onRestart() — Called when the activity has been stopped and is restarting again


Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

File: 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");
    }
```

```
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");
}
```

```
    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");
    }
}
```