ANDROID ACTIVITY LIFECYCLE

- The activity lifecycle is the set of states an activity can be in during its entire lifetime, from the time it's created to when it's destroyed and the system reclaims its resources.
- 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.
- Any activity has its distinct activity lifecycle. We have an activity stack that maintains all the activities in it. The most recent activity is at the top of the stack.
- The activity lifecycle shows the four fundamental states an activity goes through.

1. **Running State**
2. **Paused State**
3. **Resumed State**
4. **Stopped State**

1. **Running State**

Starting the activity or user interaction with an activity keeps the activity in a running state. The running state activity is at the top of the activity stack.

2. **Paused State**

Whenever the activity becomes out of focus, you can say that the activity is paused. In this state, the activity is active, but the user has not interacted with the activity for a long time.

3. **Resumed State**

Whenever any out-of-focus activity comes into focus, we call the activity present in the resumed state.

4. **Stopped State**

Whenever we close activity and move somewhere else, it is in the stopped state. So whenever the activity is in the stopped state, it won't be present in the activity stack anymore.

ANDROID ACTIVITY 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.

For example, good implementation of the lifecycle callbacks can help ensure that your app avoids:

- Crashing if the user receives a phone call or switches to another app while using your app.
- Consuming valuable system resources when the user is not actively using it.
- Losing the user's progress if they leave your app and return to it at a later time.
- Crashing or losing the user's progress when the screen rotates between landscape and portrait orientation.

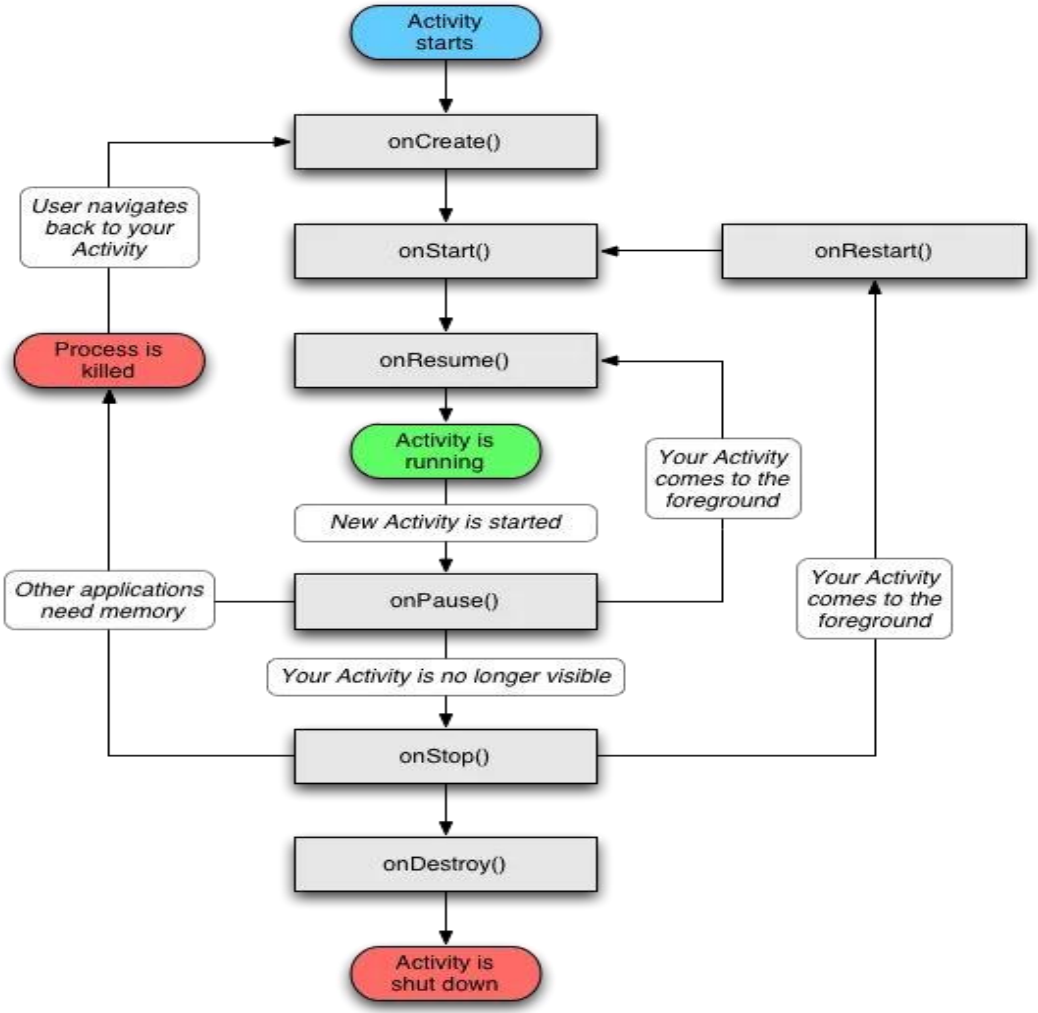
To navigate transitions between stages of the activity lifecycle, the Activity class provides a core set of seven callbacks: onCreate(), onStart(), onResume(), onPause(), onStop(), onRestart() and onDestroy(). The system invokes each of these callbacks as an activity enters a new state.

ANDROID ACTIVITY LIFECYCLE

- Android Activity Lifecycle is controlled by 7 methods of android.app.Activity class.
- The android Activity is the subclass of ContextThemeWrapper class.
- 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.
- The 7 lifecycle method of Activity describes how activity will behave at different states.

Callback 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 starts
onCreate()
User navigates back to your Activity
onStart()
onRestart()
Process is killed
onResume()
Activity is running
Your Activity comes to the foreground
New Activity is started
Other applications need memory
onPause()
Your Activity comes to the foreground
Your Activity is no longer visible
onStop()
onDestroy()
Activity is shut down

WHAT IS AN ACTIVITY IN ANDROID?

- Activity is the single screen that is available to any user whenever he starts the application.
- This screen allows the user to interact with his application.
- There can be several such screens in one application.
- Every activity consists of UI elements that help the user find out the task he wants to perform.
- Each activity has a specific lifecycle and is independent of other activities.
- In simple terms, you can think of an activity as a layout that provides the user interface.

WHAT IS AN ACTIVITY(SCREEN) IN ANDROID?

Activity can be of two types:

- **1. Main Activity** – This is the activity that loads first when the application is started.
- **2. Child Activity** – This activity is accessible from the primary(or main) activity or other child activities.

Uses of Android Activities

There are several uses of an activity which are listed as follows:

1.User Interaction : From the activity, a user can easily interact with the application.

2.Displays Data : An activity can display essential data to the user whenever required.

3.Takes input : Whenever you need to take input from the user, then you can use the activity.

4. Control : From the activity, the user can control his application and can perform his tasks.

INTRODUCTION TO ANDROID SDK

- Android development starts with the Android SDK (Software Development Kit). While there are many different programming languages and a host of IDEs (Integrated Development Environments) you can use to create an app, the SDK is a constant
- The Android SDK is what is used in Android application development. This software development kit was launched by Google so that third parties would be able to participate in designing and developing applications and software for the Android platform. The Android SDK is very much like the iPhone SDK.

Exploring the development environment

To develop Android applications, you need to have the following software installed on your computer:

- **The Java Development Kit (JDK)** Version 5 or 6, available for download.
- A compatible Java IDE such as Eclipse along with its JDT plug-in, available for download.
- The Android SDK, tools and documentation, available for download .
- The Android Development Tools (ADT) plug-in for Eclipse, available for download through the Eclipse software update mechanism.

AVD manager

The AVD Manager provides a graphical user interface in which you can create and manage Android Virtual Devices (AVDs), which are required by the Android Emulator.

Activity Life cycle program

MainActivity.java

```
package com.example.myapp;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
public class MainActivity extends AppCompatActivity {
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Log.d("lifecycle","onCreate invoked");
    Toast.makeText(getApplicationContext(),"onCreate invoked",Toast.LENGTH_LONG).show();
  }
```

Activity Life cycle program

```
@Override
 protected void onStart() {
    super.onStart();
    Log.d("lifecycle","onStart invoked");
    Toast.makeText(getApplicationContext(),"onStart invoked",Toast.LENGTH_LONG).show();
 }
 @Override
 protected void onResume() {
    super.onResume();
    Log.d("lifecycle","onResume invoked");
    Toast.makeText(getApplicationContext(),"onResume invoked",Toast.LENGTH_LONG).show();
 }
 @Override
 protected void onPause() {
    super.onPause();
    Log.d("lifecycle","onPause invoked");
    Toast.makeText(getApplicationContext(),"onPause invoked",Toast.LENGTH_LONG).show();
 }
```

Activity Life cycle program

```
@Override
  protected void onStop() {
     super.onStop();
     Log.d("lifecycle","onStop invoked");
     Toast.makeText(getApplicationContext(),"onStop invoked",Toast.LENGTH_LONG).show();
  }


@Override
  protected void onRestart() {
     super.onRestart();
     Log.d("lifecycle","onRestart invoked");
     Toast.makeText(getApplicationContext(),"onRestart invoked",Toast.LENGTH_LONG).show();
  }

  @Override
  protected void onDestroy() {
     super.onDestroy();
     Log.d("lifecycle","onDestroy invoked");
     Toast.makeText(getApplicationContext(),"onDestroy invoked",Toast.LENGTH_LONG).show();
  }
}
```

IMPORTANT LINKS

Official Site : https://developer.android.com/

https://www.tutorialspoint.com/android/android_overview.htm

https://www.geeksforgeeks.org/android-architecture/

https://www.slideshare.net/atulpanjwani5/android-final-43539730?from_action=sa ve

THANK YOU !!!