Name:- Prashant Bhosale
Roll No:- 804

EXPERIMENT 2

AIM : To study Activity LifeCycle in Android

THEORY:

There are 7 major callbacks methods or events involved in activity lifecycle.

onCreate() - This is the first callback and called when the activity is first created.

onStart() - This callback is called when the activity becomes visible to the user.

onResume() - This is called when the user starts interacting with the application.

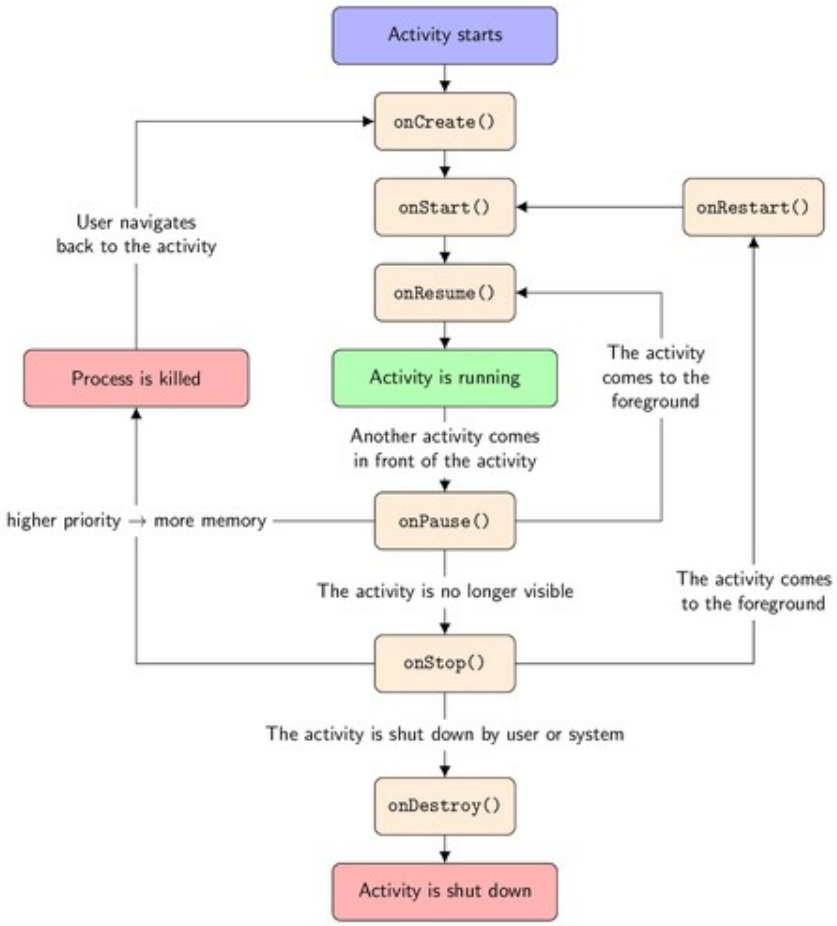
onPause() - The paused activity does not receive user input and cannot execute any code and called when the current activity is being paused and the previous activity is being resumed.

onStop() - This callback is called when the activity is no longer visible.

onDestroy() - This callback is called before the activity is destroyed by the system.

onRestart() - This callback is called when the activity restarts after stopping it.

Name:- Prashant Bhosale
Roll No:- 804



There are four process states.

S.No.	Process State
1.	Empty
2.	Foreground
3.	Background
4.	Pause

Logging options for Android Application

The Android SDK includes a useful logging utility class called **android.util.Log**. There are five type of logging message. Each type of logging message has its own method. Simply call the method and a log message is created. The message types, and their related method calls are:

Name:- Prashant Bhosale
Roll No:- 804

- The Log.e() method is used to log **errors**.
- The Log.w() method is used to log **warnings**.
- The Log.i() method is used to log **informational messages**.
- The Log.d() method is used to log **debug messages**.
- The Log.v() method is used to log **verbose messages**.

First declare a TAG constant in your class to use in the first parameter. For example, you might create an information log message as follows:

Example:

```
private static final String TAG = "MyActivity";
```

Log Message Format

Example:

Log.d(TAG, message);

The log message format is

date time PID-TID/package priority/tag: message

For example, the following log message has a priority of V and a tag of AuthZen:

```
12-10 13:02:50.071 1901-4229/com.google.android.gms V/AuthZen: Handling delegate intent.
```

PID stands for process identifier and TID is thread identifier; they can be the same if there's only one thread.

Configure logcat and monitor the log information

ASSIGNMENT

Write a program that demonstrates the complete activity lifecycle. Display the activity life cycle in logcat.

CODE:

MainActivity.java

package com.example.mca1704.activitylifecycle;

Name:- Prashant Bhosale
Roll No:- 804

```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
public class MainActivity extends AppCompatActivity {
     String my_tag = "Message";
     @Override
     protected void onCreate(Bundle savedInstanceState) {
          super.onCreate(savedInstanceState);
          setContentView(R.layout.activity_main);
          Log.i(my_tag,"OnCreated Invoked");
     }
     protected void onStart()
     {
          super.onStart();
          Log.i(my_tag,"OnStart Invoked");
     }
     protected void  onResume()
     {
          super.onResume();
          Log.i(my_tag,"OnResume Invoked");
     }
     protected void onPause()
     {
          super.onPause();
          Log.i(my_tag,"OnPause invoked");
     }
     protected void onStop()
     {
          super.onStop();
          Log.i(my_tag,"OnStop Invoked");
     }
     protected void onRestart()
     {
          super.onRestart();
          Log.i(my_tag,"OnRestart Invoked");
     }
     protected void onDestroy()
     {
          super.onDestroy();
          Log.i(my_tag,"OnDestroy Invoked");
     }
}
```

Activity_main

<?xml version="1.0" encoding="utf-8"?>

```
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
```

Name:- Prashant Bhosale
Roll No:- 804

```
    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=".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>
```

OUTPUT:

07-19 12:48:59.441 2453-2453/com.example.mca1704.activitylifecycle I/Message: OnCreated Invoked

07-19 12:48:59.441 2453-2453/com.example.mca1704.activitylifecycle I/Message: OnStart Invoked

OnResume Invoked

07-19 12:52:20.701 2453-2453/com.example.mca1704.activitylifecycle I/Message: OnPause invoked

07-19 12:52:21.781 2453-2453/com.example.mca1704.activitylifecycle I/Message: OnStop Invoked

OnDestroy Invoked

07-19 12:53:11.871 2453-2453/com.example.mca1704.activitylifecycle I/Message: OnCreated Invoked

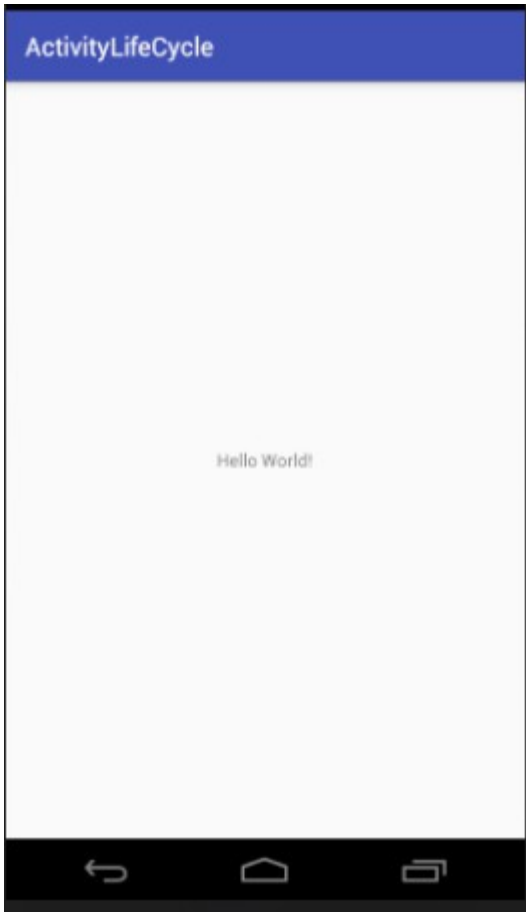
OnStart Invoked

OnResume Invoked

07-19 12:53:21.051 2453-2453/com.example.mca1704.activitylifecycle I/Message: OnPause invoked

07-19 12:53:21.781 2453-2453/com.example.mca1704.activitylifecycle I/Message: OnStop Invoked

Name:- Prashant Bhosale
Roll No:- 804



CONCLUSION:

In this practical we have studied Activity LifeCycle in Android