Android Activity Lifecycle methods

**1.** In this lab we will be learning how to use and extend the Android user interface library.

a. Views, View Groups, Layouts, and Widgets are and how they relate to each other.

b. How to declare and reference resources in code.

c. How to navigate between multiple activities.

d. How to share the data between the activities.

e. Explore life-cycle methods of an activity.

f. How to use Events and Event Listeners.

g. How to create Toast Notifications.

Code:

**activity\_main.xml**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.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="com.example.helloworld.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" />  
</androidx.constraintlayout.widget.ConstraintLayout>

**MainActivity.java**

package com.example.helloworld;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
//import android.os.Bundle;  
//import android.app.Activity;  
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");  
 }  
 @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");  
 }  
}

Program 2

2. You will expand on your knowledge of the Android user interface library.

a. How to declare layouts statically as an xml resource.

b. How to create custom Views from scratch to suit a specific need.

c. How to create Options and Context Menus.

d. How to use ListAdapter and ArrayAdapter to bind data source to a List View.

e. How to create AlertDialog and progress Dialog in your activity.

Android Widgets

There are given a lot of **android widgets** with simplified examples such as Button, EditText, AutoCompleteTextView, ToggleButton, DatePicker, TimePicker, ProgressBar etc.

Android widgets are easy to learn. The widely used android widgets with examples are given below:

**[Android Button](https://www.javatpoint.com/android-working-with-button)**

Let's learn how to perform event handling on button click.

**[Android Toast](https://www.javatpoint.com/android-toast-example)**

Displays information for the short duration of time.

**[Custom Toast](https://www.javatpoint.com/android-custom-toast-example)**

We are able to customize the toast, such as we can display image on the toast

**[ToggleButton](https://www.javatpoint.com/android-togglebutton-example)**

It has two states ON/OFF.

**[CheckBox](https://www.javatpoint.com/android-checkbox-example)**

Let's see the application of simple food ordering.

**[AlertDialog](https://www.javatpoint.com/android-alert-dialog-example)**

AlertDialog displays a alert dialog containing the message with OK and Cancel buttons.

**[Spinner](https://www.javatpoint.com/android-spinner-example)**

Spinner displays the multiple options, but only one can be selected at a time.

**[AutoCompleteTextView](https://www.javatpoint.com/android-autocompletetextview-example)**

Let's see the simple example of AutoCompleteTextView.

**[RatingBar](https://www.javatpoint.com/android-rating-bar-example)**

RatingBar displays the rating bar.

**[DatePicker](https://www.javatpoint.com/android-datepicker-example)**

Datepicker displays the datepicker dialog that can be used to pick the date

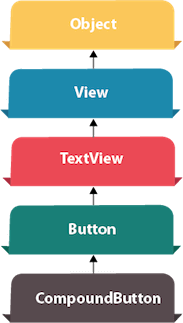
**T[imePicker](https://www.javatpoint.com/android-timepicker-example)**

TimePicker displays the timepicker dialog that can be used to pick the time.

**[ProgressBar](https://www.javatpoint.com/android-progressbar-example)**

ProgressBar displays progress task.

Android Button Example



Android Button represents a push-button. The android.widget.Button is subclass of TextView class and CompoundButton is the subclass of Button class.

There are different types of buttons in android such as RadioButton, ToggleButton, CompoundButton etc.

Android Button Example with Listener

Here, we are going to create two textfields and one button for sum of two numbers. If user clicks button, sum of two input values is displayed on the Toast.

We can perform action on button using different types such as calling listener on button or adding onClick property of button in activity's xml file.

button.setOnClickListener(new View.OnClickListener() {

           @Override

            public void onClick(View view) {

               //code

            }

});

**<Button**

        android:onClick="methodName"

**/>**

Code:

**activity\_main.xml**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.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=".MainActivity"  
 tools:visibility="visible">  
  
 <Button  
 android:id="@+id/button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Long Click on Me"  
 android:textSize="25sp"  
 android:visibility="visible"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**MainActivity.java**

**package com.example.contextmenu;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.view.ContextMenu;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.Button;  
import android.widget.Toast;  
  
public class MainActivity extends AppCompatActivity {  
  
 private Button button;  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 button = findViewById(R.id.*button*);  
 registerForContextMenu(button);  
  
  
 }  
  
 @Override  
 public void onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo) {  
getMenuInflater().inflate(R.menu.*menu*,menu);  
 super.onCreateContextMenu(menu, v, menuInfo);  
 }  
  
 @Override  
 public boolean onContextItemSelected(@NonNull MenuItem item) {  
 Toast.*makeText*(this, ""+item.getTitle(), Toast.*LENGTH\_SHORT*).show();  
 return super.onContextItemSelected(item);  
 }  
}**

**menu.xml**

**<?xml version="1.0" encoding="utf-8"?>  
<menu xmlns:android="http://schemas.android.com/apk/res/android">  
  
 <item  
 android:title="@string/one"  
 />  
 <item  
 android:title="@string/two"  
 />  
 <item  
 android:title="@string/three"  
 />  
 <item  
 android:title="@string/four"  
 />  
 <item  
 android:title="@string/five"  
 />  
</menu>**