

Unit-4: Android Activity

Syllabus:

Unit -4

Android Activity [4 HRS]

The Activity life cycle, Creating multiple activities, Declaring activities in the manifest, Connecting activities with intents, Passing data between activities, Getting a result back from a child activity, Getting and setting data to/from the layout file.

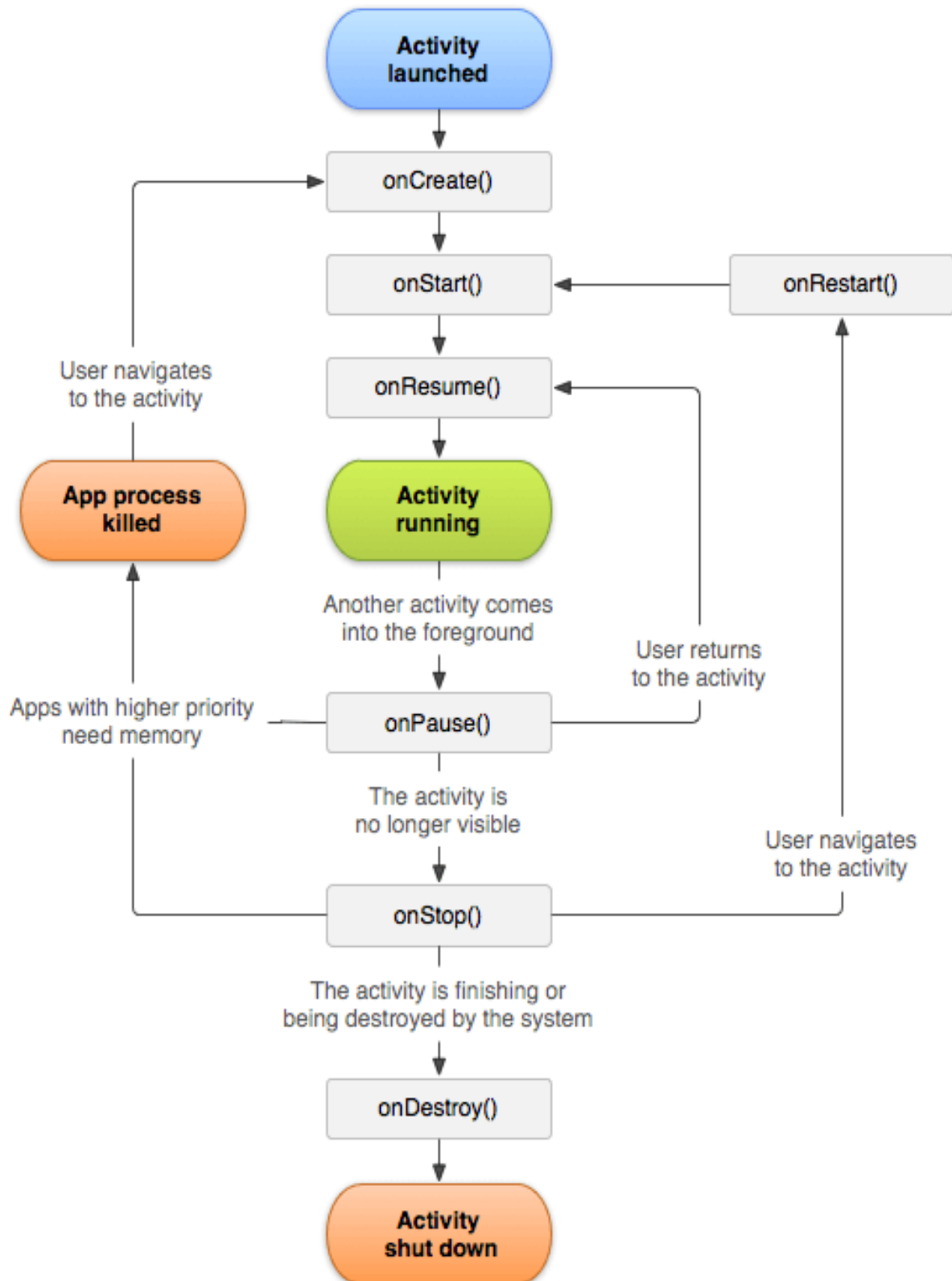
Activity lifecycle:

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.

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.



onCreate()

You must implement this callback, which fires when the system first creates the activity. On activity creation, the activity enters the Created state. In the onCreate() method, you perform basic application startup logic that should happen only once for the entire life of the activity.

```
protected void onCreate(Bundle b)
{
    super.onCreate(b);
    setContentView(R.layout.activity_life_cycle);
    Log.d("LifeCycle", "onCreate Invoked");
}
```

onStart()

When the activity enters the Started state, the system invokes this callback. The onStart() call makes the **activity visible to the user**, as the app prepares for the activity to enter the foreground and become interactive. For example, this method is where the app initializes the code that maintains the UI.

When the activity moves to the started state, any lifecycle-aware component tied to the activity's lifecycle will receive the ON START event.

The onStart() method completes very quickly and, as with the Created state, the activity does not stay resident in the Started state. Once this callback finishes, the activity enters the Resumed state, and the system invokes the onResume() method.

```
protected void onStart()
{
    super.onStart();
    Log.d("LifeCycle", "onStart Invoked");
}
```

onResume()

When the activity enters the Resumed state, it comes to the foreground, and then the system invokes the onResume() callback. This is the state in which **the app interacts with the user**. The app stays in this state until something happens to take focus away from the app. Such an event might be, for instance, receiving a phone call, the user's navigating to another activity, or the device screen's turning off.

```
protected void onResume()
{
    super.onResume();
    Log.d("LifeCycle", "onResume Invoked");
}
```

onPause()

The system calls this method as the first indication that the user is leaving your activity (though it does not always mean the activity is being destroyed); it indicates that the activity is no longer in the foreground (though it may still be visible if the user is in multi-window mode). Use the `onPause()` method to pause or adjust operations that should not continue (or should continue in moderation) while the Activity is in the Paused state, and that you expect to resume shortly. There are several reasons why an activity may enter this state. For example:

- Some event interrupts app execution, as described in the `onResume()` section. This is the most common case.
- In Android 7.0 (API level 24) or higher, multiple apps run in multi-window mode. Because only one of the apps (windows) has focus at any time, the system pauses all of the other apps.
- A new, semi-transparent activity (such as a dialog) opens. As long as the activity is still partially visible but not in focus, it remains paused.

```
protected void onPause()
{
    super.onPause();
    Log.d("LifeCycle", "onPause Invoked");
}
```

onStop()

When your activity is no longer visible to the user, it has entered the Stopped state, and the system invokes the `onStop()` callback. This may occur, for example, when a newly launched activity covers the entire screen. The system may also call `onStop()` when the activity has finished running, and is about to be terminated.

```
protected void onStop()
{
    super.onStop();
    Log.d("LifeCycle", "onStop Invoked");
}
```

onDestroy()

`onDestroy()` is called before the activity is destroyed. The system invokes this callback either because:

- the activity is finishing (due to the user completely dismissing the activity or due to `finish()` being called on the activity), or
- the system is temporarily destroying the activity due to a configuration change (such as device rotation or multi-window mode)
- When the activity moves to the destroyed state, any lifecycle-aware component tied to the activity's lifecycle will receive the ON_DESTROY event. This is where the lifecycle components can clean up anything it needs to before the Activity is destroyed.

Following is the example of activity life cycle of an android application

activity life_cycle.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"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Welcome to Activity Life Cycle"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.54"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.127" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

ActivityLifeCycle.java

```
package com.example.androidpractice;

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

public class ActivityLifeCycle extends Activity {
    protected void onCreate(Bundle b)
    {
        super.onCreate(b);
        setContentView(R.layout.activity_life_cycle);
        Log.d("LifeCycle", "onCreate Invoked");
    }
    protected void onStart()
    {
        super.onStart();
        Log.d("LifeCycle", "onStart Invoked");
    }
    protected void onResume()
    {
        super.onResume();
    }
}
```

```
        Log.d("LifeCycle","onResume Invoked");
    }
    protected void onPause()
    {
        super.onPause();
        Log.d("LifeCycle","onPouse Invoked");
    }
    protected void onStop()
    {
        super.onStop();
        Log.d("LifeCycle","onStop Invoked");
    }
    protected void onRestart()
    {
        super.onRestart();
        Log.d("LifeCycle","onRestart Invoked");
    }
    protected void onDestroy()
    {
        super.onDestroy();
        Log.d("LifeCycle","onDestroy Invoked");
    }
}
```

Run this code in emulator and analyze the result in the Logcat.



```
Logcat
Emulator Testing_Phone Android com.example.androidpractice (140) Verbose
2022-03-10 17:10:38.404 14096-14096/? I/art: Late-enabling -Xcheck:jni
2022-03-10 17:10:38.512 14096-14096/com.example.androidpractice W/System: ClassLoader referenced unk
2022-03-10 17:10:38.749 14096-14096/com.example.androidpractice D/Lifecycle: onCreate Invoked
2022-03-10 17:10:38.754 14096-14096/com.example.androidpractice D/Lifecycle: onStart Invoked
2022-03-10 17:10:38.755 14096-14096/com.example.androidpractice D/Lifecycle: onResume Invoked
2022-03-10 17:10:38.822 14096-14119/com.example.androidpractice D/OpenGLRenderer: Use EGL_SWAP_BEHAV
```

Here you can see, onCreate Invoked, onStartInvoked, onResume Invoked

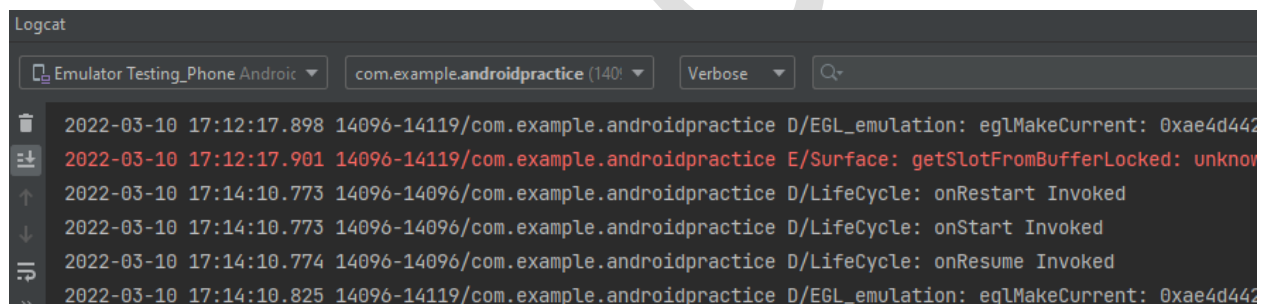
Now press on home button and see in Logcat



```
Logcat
Emulator Testing_Phone Android com.example.androidpractice (140) Verbose
2022-03-10 17:10:39.027 14096-14119/com.example.androidpractice E/eglCodecCommon: glUtilsParamSize: unknow
2022-03-10 17:10:39.087 14096-14119/com.example.androidpractice D/EGL_emulation: eglMakeCurrent: 0xae4d44
2022-03-10 17:12:17.841 14096-14096/com.example.androidpractice D/Lifecycle: onPause Invoked
2022-03-10 17:12:17.868 14096-14096/com.example.androidpractice D/Lifecycle: onStop Invoked
2022-03-10 17:12:17.898 14096-14119/com.example.androidpractice D/EGL_emulation: eglMakeCurrent: 0xae4d44
```

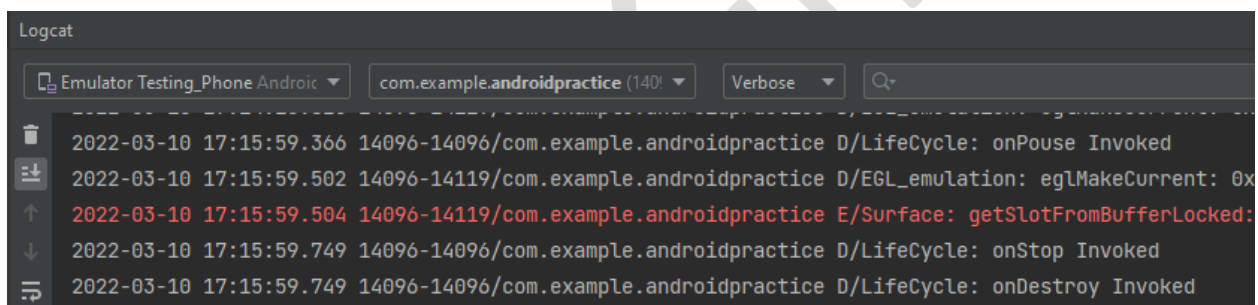
Here you can see, onPause Invoked, onStop Invoked

Now press in the Munu button and click on application



Again the activity is Restarted, you can see, onRestart Invoked, onStart Invoked, onResume Invoked

Now click on back button



Now you can see, onPause is Invoked for a while and onStop Invoked then onDestroy is Invoked.

Now the activity life cycle is permanently destroyed

Creating Multiple Activities:

We can create multiple activities in the application and can link those activities as required. Let's create two activities "FirstActivity" and "SecondActivity" with some widgets like TextView and Button.

Activity_main.xml

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

<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="This is the First Activity" />

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Go to Second Activity" />

</LinearLayout>
```

Activity_main2.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="This is the Second Activity" />
```

```
<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Go to First Activity" />
</LinearLayout>
```

MainActivity.java

```
package com.example.unit4_lab1;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.content.Intent;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button button1 = findViewById(R.id.button1);
        button1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this, MainActivity2.class);
                startActivity(intent);
            }
        });
    }
}
```

MainActivity2.java

```
package com.example.unit4_lab1;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.content.Intent;
import android.view.View;
import android.widget.Button;

public class MainActivity2 extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);
        Button button2 = findViewById(R.id.button2);
        button2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity2.this, MainActivity.class);
                startActivity(intent);
            }
        });
    }
}
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.unit4_lab1">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.Unit4_lab1">
        <activity
            android:name=".MainActivity2"
            />
    </application>
</manifest>
```

```
<activity
    android:name=".MainActivity"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
</application>

</manifest>
```

Output:



Declaring Activities in Manifest file:

Each android application you build will include a file called *AndroidManifest.xml* which is placed in the root of the project hierarchy. It lets you define the structure and metadata of your android application and its components.

You have to declare your each and every class in android manifest file, so that it recognizes them as the Activity. So after the end of the creating Activity you have to define activity in manifest file.

When declaring an activity (an Activity subclass) that implements part of the application's visual user interface. All activities must be represented by **<activity>** elements in the manifest file. Any that are not declared there will not be seen by the system and will never be run.

The Activity which you want of lunch first while app is running, then you have to mention that Activity inside the **<intent-filter>** element of the manifest file which is placed inside the **<activity>** element.

All other Activities are place as separate activity without **<intent-filter>** element.

Following code shows the declaration of two activities **"FirstActivity"** and **"SecondActivity"** as we created earlier.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.androidpractice">
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.AndroidPractice">
        <activity
            android:name=".FirstActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <activity android:name=".SecondActivity"
            android:exported="true">
        </activity>

    </application>
</manifest>
```

Here, we want to lunch **FirstActivity** while running this app, so I declare **<intent-filter>** element inside that activity mention the action as MAIN and category as LAUNCHER.

Connecting Activity with Intent:

Android Intent is the *message* that is passed between components such as activities, content providers, broadcast receivers, services etc. It is generally used with startActivity() method to invoke activity, broadcast receivers etc.

***Note:** The dictionary meaning of intent is intention or purpose. So, it can be described as the intention to do action.*

Android intents are mainly used to:

- Start the service
- Launch an activity
- Display a web page
- Display a list of contacts
- Broadcast a message
- Dial a phone call etc.

There are two types of intents in android: implicit and explicit.

1) Implicit Intent:

Implicit Intent doesn't specify the component. In such case, intent provides information of available components provided by the system that is to be invoked. For example, you may write the following code to view the webpage.

```
Intent intent=new Intent (Intent.ACTION_VIEW);
intent.setData(Uri.parse("http://www.google.com"));
startActivity(intent);
```

Example: Following example shows the visit the desired webpage by entering url in edit text box by using implicit intent.

implicit_intent.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Enter Your Url below"
    android:layout_gravity="center"
    android:layout_marginTop="40dp"/>

<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/urlid"
    android:layout_gravity="center"
    android:hint="Entr your site url"/>

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Visit"
    android:id="@+id/visitbtnid"
    android:layout_gravity="center"/>

</LinearLayout>
```

ImplicitIntent.java

```
package com.example.androidpractice;

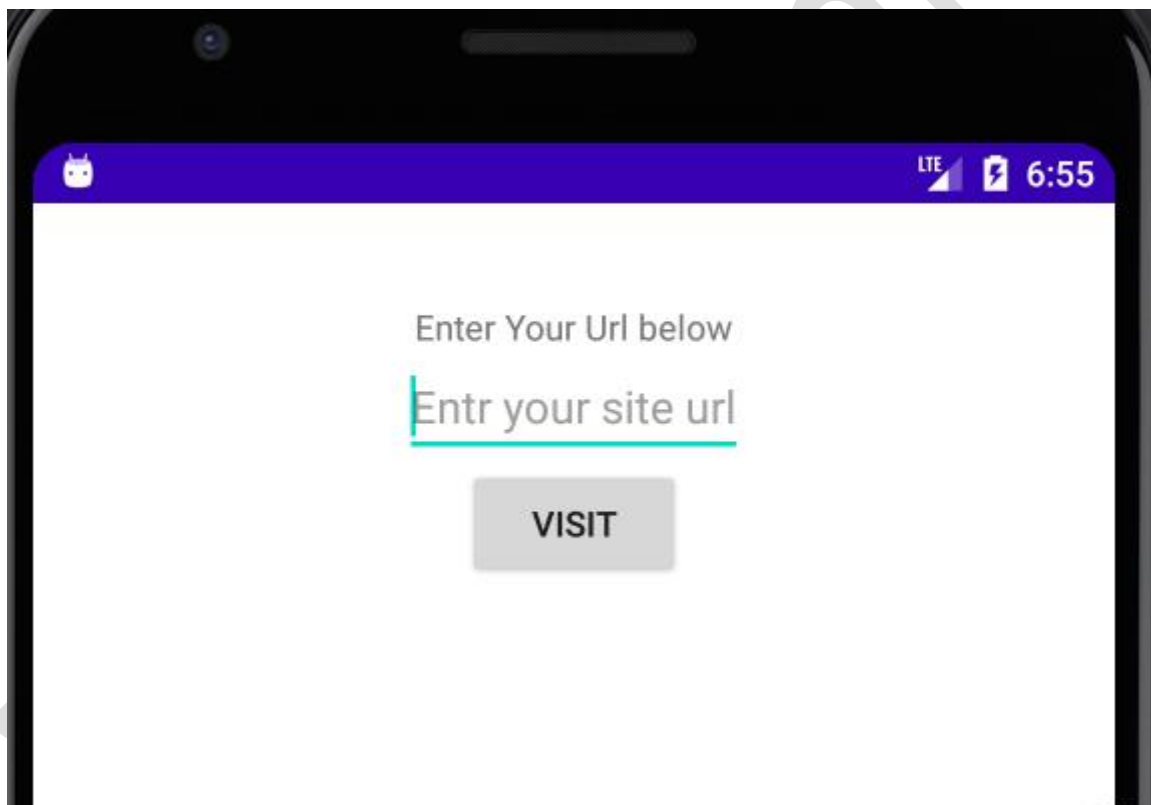
import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class ImplicitIntent extends Activity {
    protected void onCreate(Bundle b)
    {
        super.onCreate(b);
        setContentView(R.layout.implicit_intent);

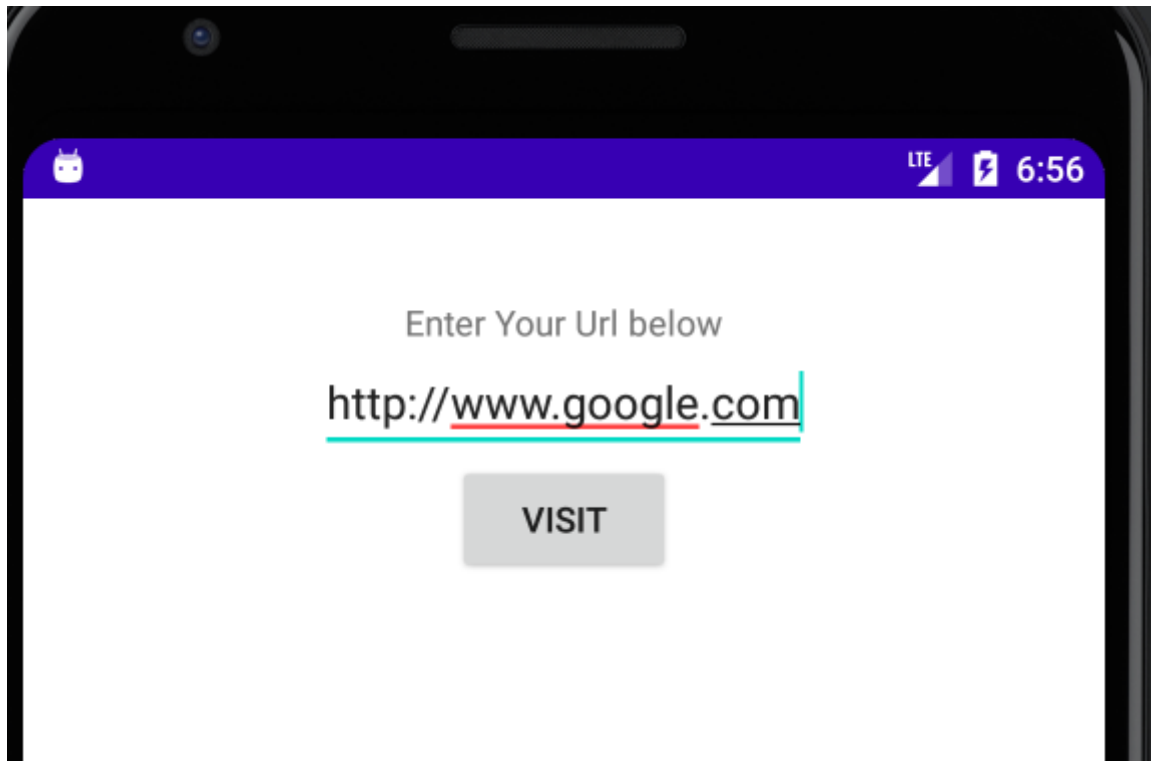
        EditText enteredurl = findViewById(R.id.urlid);
        Button visitbtn = findViewById(R.id.visitbtnid);
```



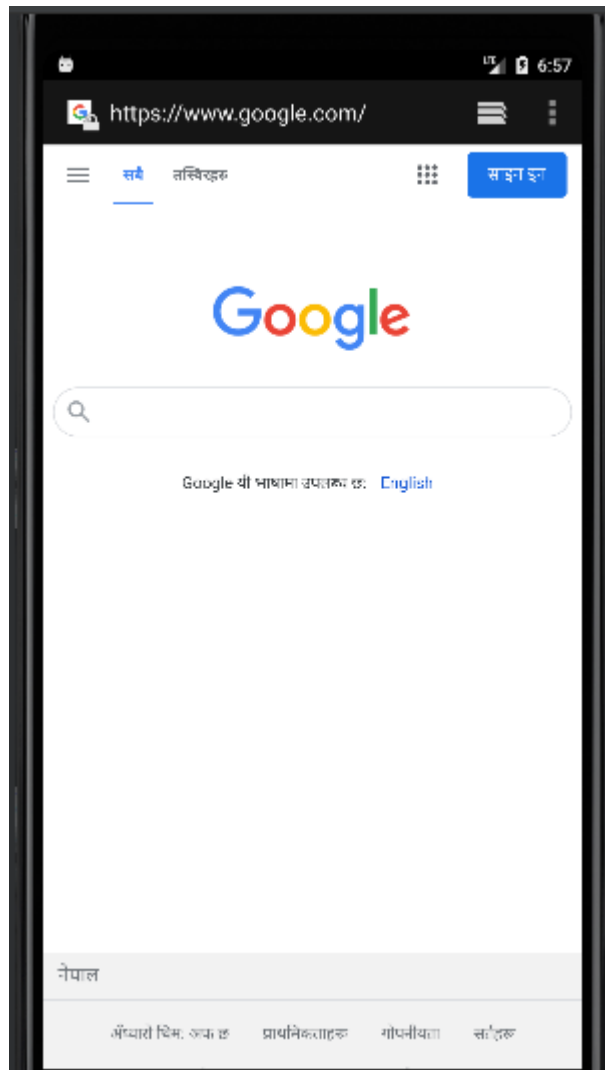
```
visitbtn.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        String myurl = enteredurl.getText().toString();  
        Intent i = new Intent(Intent.ACTION_VIEW);  
        i.setData(Uri.parse(myurl));  
        startActivity(i);  
    }  
});  
}  
}
```



Enter your desired site url and press in visit button



After pressing **VISIT** button following Google web page will display.



2) *Explicit Intent*:

Explicit Intent specifies the component. In such case, intent provides the external class to be invoked. Which is use to navigate in to the different activity page of same application.

```
Intent intent = new Intent (getApplicationContext(), ActivityTwo.class);
startActivity(intent);
```

Example: In this example we create two activities “**FirstActivity**” and “**SecondActivity**” then link both using intent.

first_activity.xml

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

```
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical">
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Welcome to First Page"
    android:layout_gravity="center"
    android:layout_marginTop="20dp"/>
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Go to Next Page"
    android:id="@+id/fbtn1"
    android:layout_gravity="center"
    android:layout_marginTop="40dp"/>
</LinearLayout>
```

second_activity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Congratulations !!! you reached in second page"
        android:layout_gravity="center"
        android:layout_marginTop="20dp"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Thank Your !!!"
        android:layout_gravity="center"
        android:layout_marginTop="40dp"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Go back to previous page"
        android:id="@+id/sbtn1"
```

```
        android:layout_gravity="center"/>
</LinearLayout>
```

FirstActivity.java

```
package com.example.androidpractice;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class FirstActivity extends Activity {
    protected void onCreate(Bundle b)
    {
        super.onCreate(b);
        setContentView(R.layout.first_activity);

        Button btn = findViewById(R.id.fbtn1);

        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(getApplicationContext(),SecondActivity.class);
                startActivity(intent);
            }
        });
    }
}
```

SecondActivity.java

```
package com.example.androidpractice;

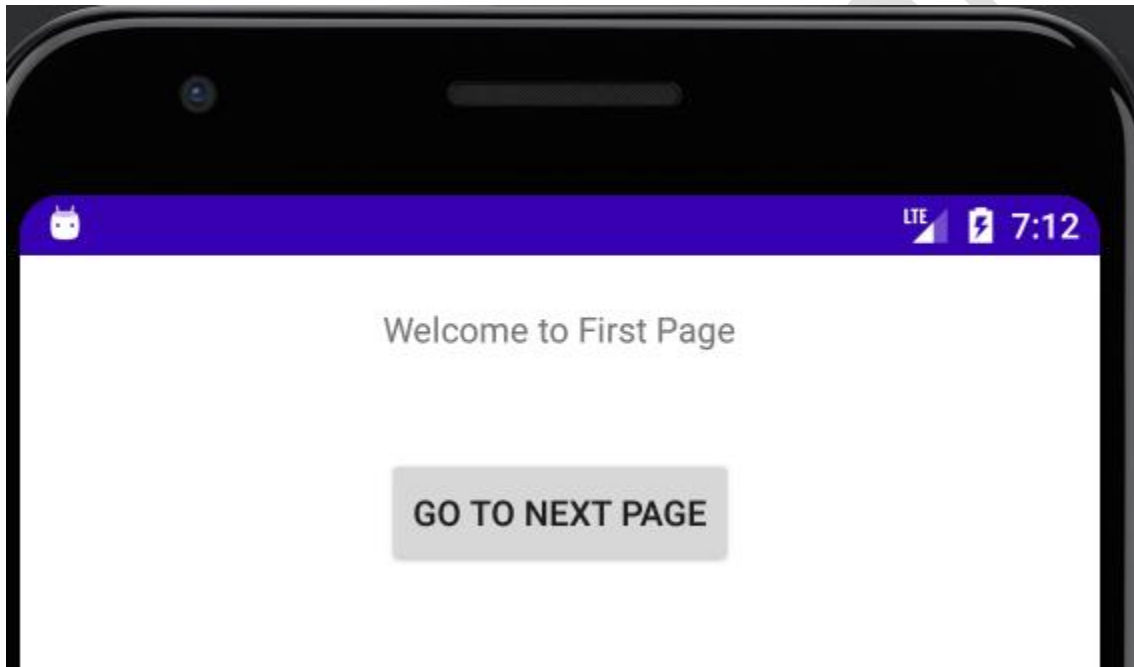
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class SecondActivity extends Activity {
    protected void onCreate(Bundle b)
    {
        super.onCreate(b);
```

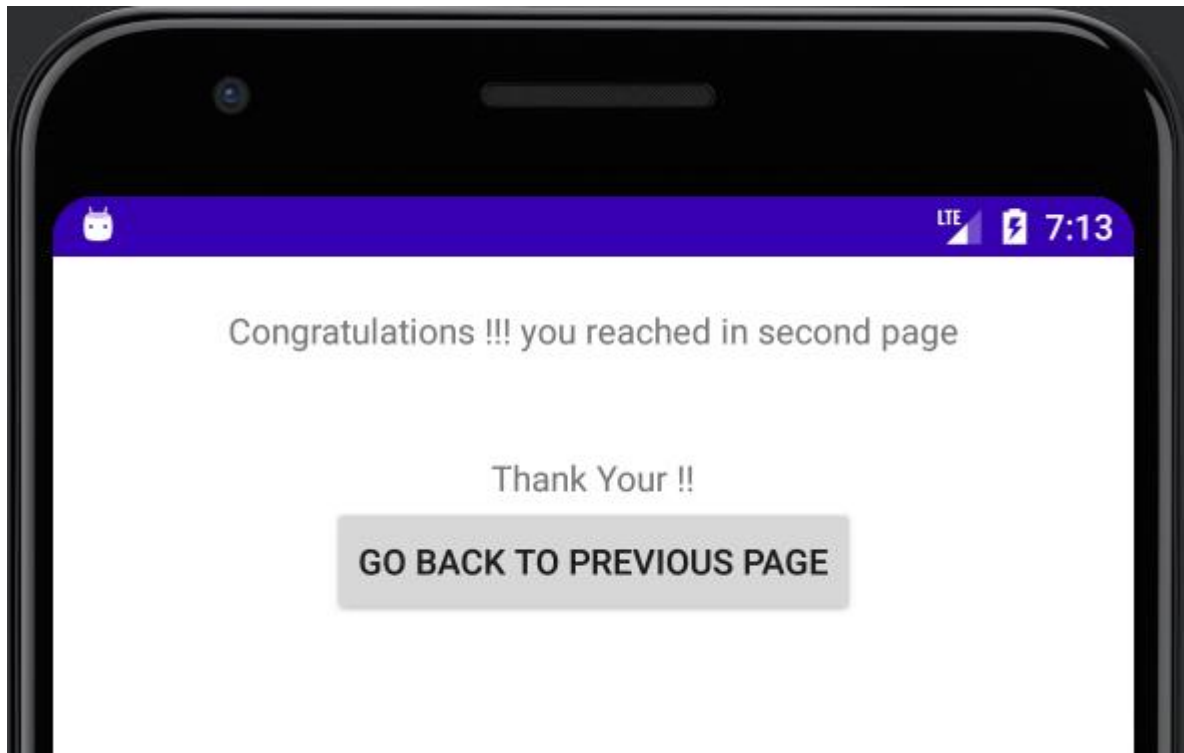
```
setContentView(R.layout.second_activity);

Button btn = findViewById(R.id.sbtn1);

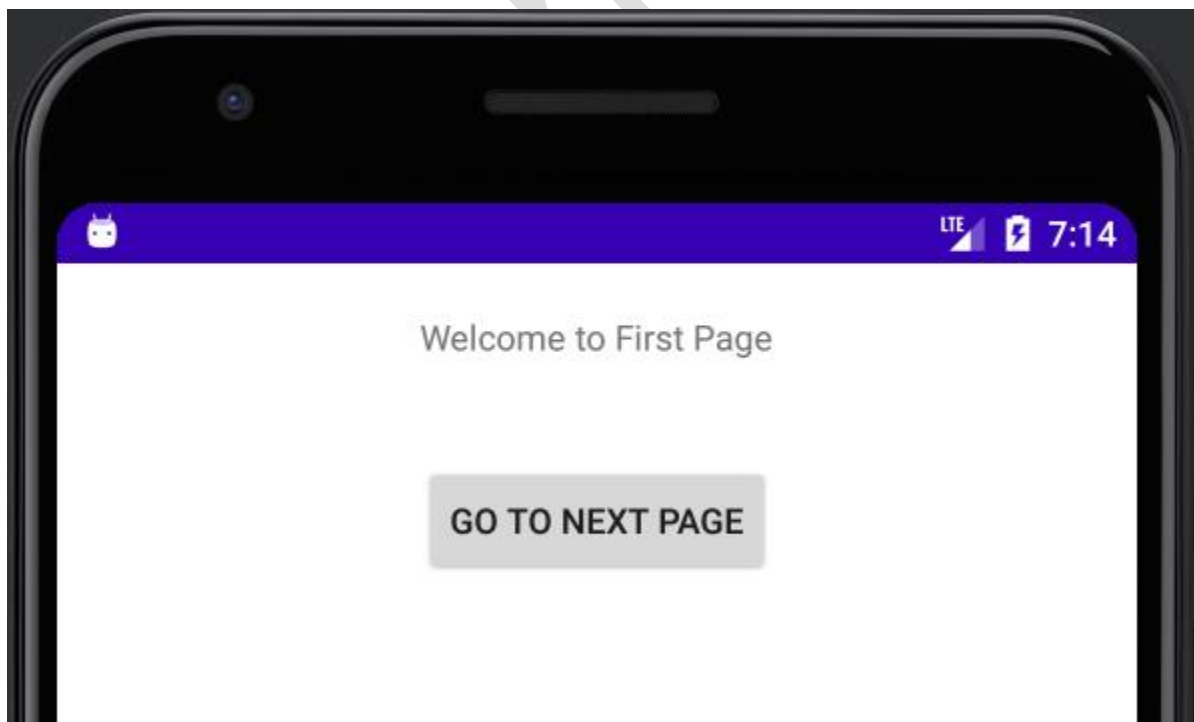
btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(getApplicationContext(),FirstActivity.class);
        startActivity(intent);
    }
});
}
```



Click on **GO TO NEXT PAGE** button



Click on **GO BACK TO PREVIOUS PAGE** button



Teksan Gharti Magar

Passing Data between Activities:

passdata.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="40dp"
        android:text="Enter your information">

    </TextView>
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:id="@+id/idname"
        android:layout_marginTop="20dp"
        android:hint="Enter Your Name">
    </EditText>
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:id="@+id/idaddr"
        android:hint="Enter Your Address">
    </EditText>
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:id="@+id/idphn"
        android:hint="Enter Your Phone Number">
    </EditText>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:id="@+id/btnsub"
        android:text="submit">
```


</Button>

</LinearLayout>

receivedata.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="40dp"
        android:text="My Information:"
        android:layout_gravity="center">
    </TextView>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="40dp"
        android:id="@+id/myinfo"
        android:layout_gravity="center">
    </TextView>
</LinearLayout>
```

PassData.java

```
package com.example.androidpractice;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class PassData extends Activity {
    protected void onCreate(Bundle b)
    {
        super.onCreate(b);
        setContentView(R.layout.passdata);
    }
}
```

```
EditText myname = findViewById(R.id.idname);
EditText myaddr = findViewById(R.id.idaddr);
EditText myphon = findViewById(R.id.idphn);
Button btnsend = findViewById(R.id.btnsub);

btnsend.setOnClickListener(new View.OnClickListener(){
    public void onClick(View v)
    {
        String sendname = myname.getText().toString();
        String sendaddr = myaddr.getText().toString();
        String sendphn = myphon.getText().toString();

        Intent intent = new Intent(getApplicationContext(),ReceiveData.class);

        intent.putExtra("name_key",sendname);
        intent.putExtra("address_key",sendaddr);
        intent.putExtra("phone_key",sendphn);

        startActivity(intent);
    }
});
}
```

ReceiveData.java

```
package com.example.androidpractice;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.widget.TextView;

public class ReceiveData extends Activity {
    protected void onCreate(Bundle b)
    {
        super.onCreate(b);
        setContentView(R.layout.receivedata);

        TextView receivedata = findViewById(R.id.myinfo);

        Intent in = getIntent();

        String name = in.getStringExtra("name_key");
        String address = in.getStringExtra("address_key");
        String phone = in.getStringExtra("phone_key");
    }
}
```

```
receivedata.setText("Name = "+ name + "\n Address = "+address+"\n Phone No = "+phone);  
    }  
}
```

Getting and Setting to and from the layout file:

Create an android app to add the two integer numbers enter by user.

calculator.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical">  
  
    <EditText  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:layout_marginTop="50dp"  
        android:hint="Enter first number"  
        android:id="@+id/num1">  
  
    </EditText>  
  
    <EditText  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:hint="Enter first number"  
        android:id="@+id/num2">  
  
    </EditText>  
    <Button  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="Add"  
        android:layout_gravity="center"  
        android:id="@+id/btnadd">  
  
    </Button>  
    <TextView  
        android:layout_width="wrap_content"
```

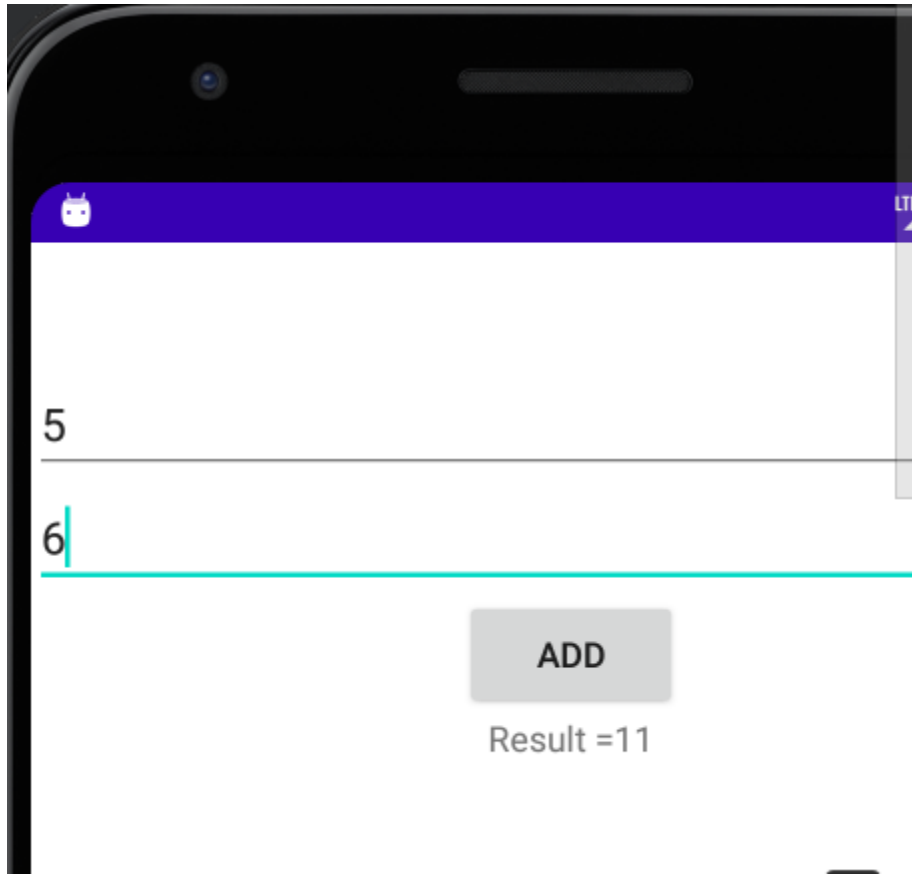
```
android:layout_height="wrap_content"  
android:layout_gravity="center"  
android:id="@+id/txtresult">
```

```
</TextView>
```

```
</LinearLayout>
```

Calculator.java

```
package com.example.androidpractice;  
  
import android.app.Activity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
  
public class Calculator extends Activity {  
    protected void onCreate(Bundle b)  
    {  
        super.onCreate(b);  
        setContentView(R.layout.calculator);  
        Button btn = findViewById(R.id.btnadd);  
        EditText number1 = findViewById(R.id.num1);  
        EditText number2 = findViewById(R.id.num2);  
        TextView result = findViewById(R.id.txtresult);  
  
        btn.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View view) {  
                int n1,n2,res;  
                n1=Integer.parseInt(number1.getText().toString());  
                n2 = Integer.parseInt(number2.getText().toString());  
                res = n1+n2;  
                result.setText("Result =" +res);  
            }  
        });  
    }  
}
```



Getting the result Back from child activity

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    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"
    android:orientation="vertical"
    tools:context=".MainActivity"
    >
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/textViewDisplayMessage"
        android:gravity="center"
        android:textSize="30sp"
        android:textStyle="bold"
        android:textColor="#4CAF50"
    />
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        >
        <Button
            android:id="@+id/addButton"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="add"
            android:layout_marginLeft="70dp"
        />
        <Button
            android:id="@+id/subButton"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="sub"
            android:layout_marginLeft="40dp"
        />
    </LinearLayout>
```

```
<TextView
    android:id="@+id/result"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Result"
    android:gravity="center"
/>
```

```
</LinearLayout>
```

MainActivity.java

```
package com.example.myapplication;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    private static final int REQUEST_CODE_ADD = 1;
    private static final int REQUEST_CODE_SUB = 2;
    private TextView resultTextView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        resultTextView = findViewById(R.id.result);

        Button addBut = findViewById(R.id.addButton);
        addBut.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(MainActivity.this,
MainActivity2.class);
                startActivityForResult(i,
REQUEST_CODE_ADD);
            }
        });

        Button subBut = findViewById(R.id.subButton);
```

```
        subBut.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(MainActivity.this,
MainActivity3.class);
                startActivityForResult(i,
REQUEST_CODE_SUB);
            }
        });
    }

    @Override
    protected void onActivityResult(int requestCode, int
resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode,
data);

        if (resultCode == RESULT_OK) {
            int result = 0;
            if (data != null) {
                switch (requestCode) {
                    case REQUEST_CODE_ADD:
                        result =
data.getIntExtra("resActivity2_key", 0);
                        resultTextView.setText("Add
Result: " + result);
                        break;
                    case REQUEST_CODE_SUB:
                        result =
data.getIntExtra("resActivity3_key", 0);
                        resultTextView.setText("Sub
Result: " + result);
                        break;
                }
            }
        }
    }
}
```


Activity_main2.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    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"
    android:orientation="vertical"
    tools:context=".MainActivity2">
    <EditText
        android:id="@+id/editTextNum1InActivity2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter the first number"
    />

    <EditText
        android:id="@+id/editTextNum2InActivity2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter the second number"
    />

    <Button
        android:id="@+id/addButtonInActivity2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="add"
    />

</LinearLayout>
```

MainActivity2.java

```
package com.example.myapplication;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.content.Intent;

public class MainActivity2 extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);

        EditText num1Activity2=findViewById(R.id.editTextNum1InActivity2);
        EditText num2Activity2=findViewById(R.id.editTextNum2InActivity2);
        Button btnActivity2=findViewById(R.id.addButtonInActivity2);
        btnActivity2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                int n1,n2,resActivity2;
                n1=Integer.parseInt(num1Activity2.getText().toString());
                n2=Integer.parseInt(num2Activity2.getText().toString());
                resActivity2=n1+n2;

                Intent resultIntent = new Intent();
                resultIntent.putExtra("resActivity2_key", resActivity2);
                setResult(RESULT_OK, resultIntent);
                finish();

            }
        });

    }
}
```

Activity_main3.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    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"
    android:orientation="vertical"
    tools:context=".MainActivity3">
    <EditText
        android:id="@+id/editTextNum1InActivity3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter the first number"
    />

    <EditText
        android:id="@+id/editTextNum2InActivity3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter the second number"
    />

    <Button
        android:id="@+id/addButtonInActivity3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="sub"
    />

</LinearLayout>
```

MainActivity3.java

```
package com.example.myapplication;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
```

```
public class MainActivity3 extends AppCompatActivity {

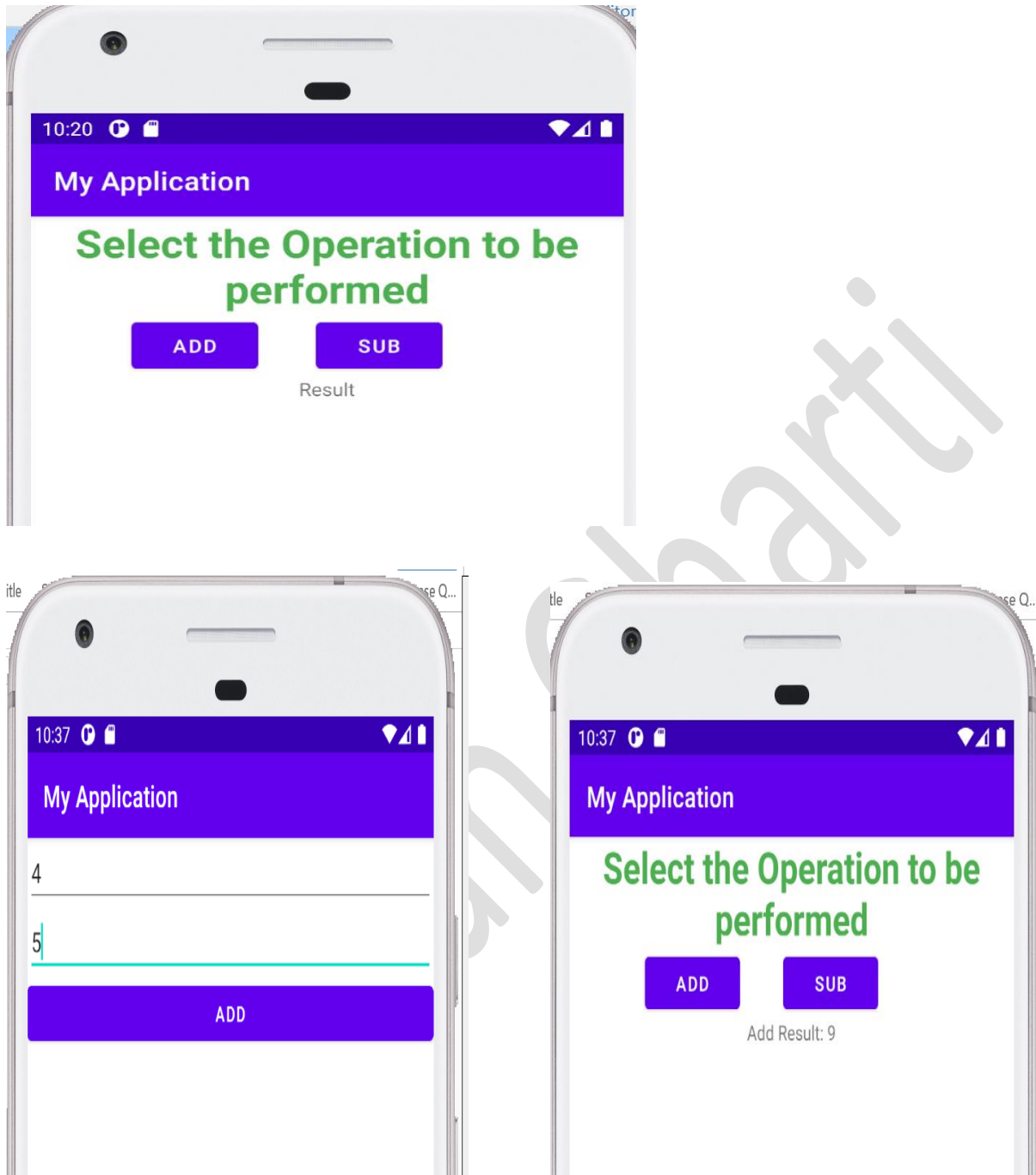
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main3);
        EditText num1Activity3=findViewById(R.id.editTextNum1InActivity3);
        EditText num2Activity3=findViewById(R.id.editTextNum2InActivity3);
        Button btnActivity3=findViewById(R.id.addButtonInActivity3);
        btnActivity3.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                int n1,n2,resActivity3;
                n1=Integer.parseInt(num1Activity3.getText().toString());
                n2=Integer.parseInt(num2Activity3.getText().toString());
                resActivity3=n1-n2;

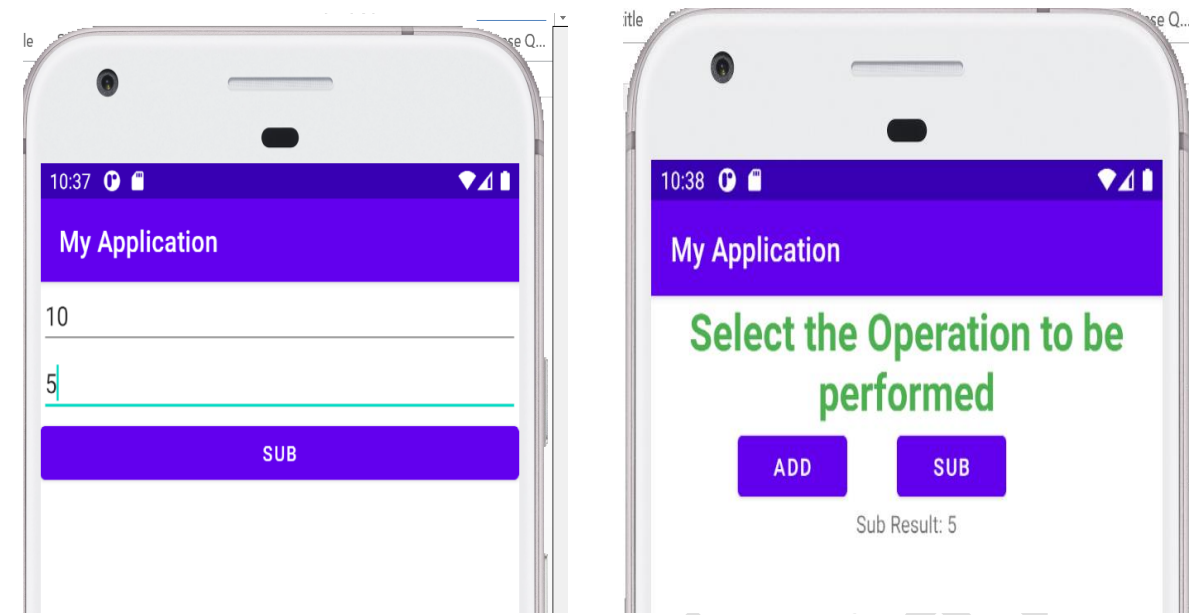
                Intent resultIntent = new Intent();
                resultIntent.putExtra("resActivity3_key", resActivity3);
                setResult(RESULT_OK, resultIntent);
                finish();
            }
        });
    }
}
```

string.xml

```
<resources>
    <string name="app_name">My Application</string>
    <string name="textViewDisplayMessage">Select the Operation to be
performed</string>
</resources>
```

Output:





End of unit-4