Unit 4

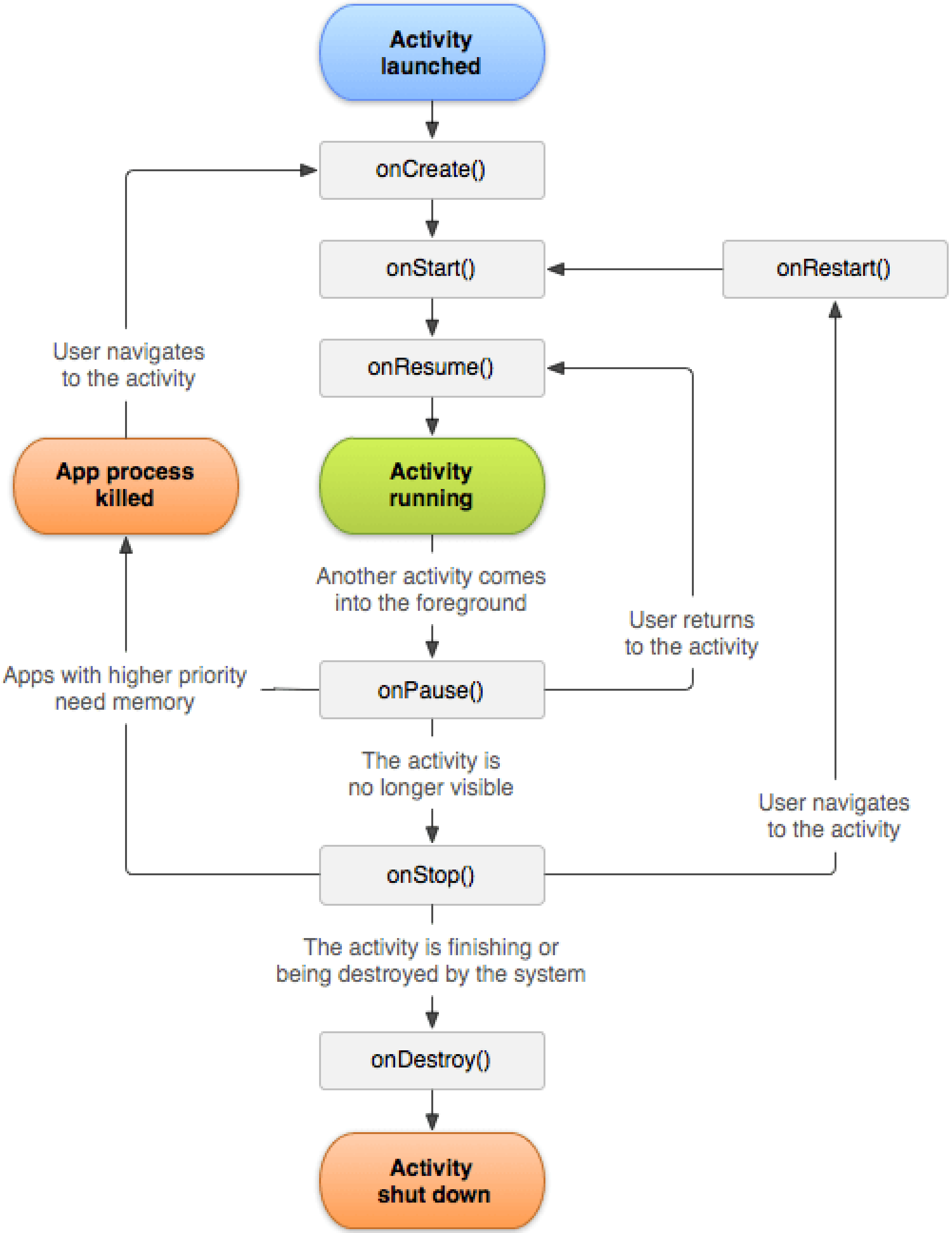
Android Activity

The Activity life cycle

As a user navigates through, out of, and back to our app, the Activity instances in our app transition through different states in their 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.

Within the lifecycle callback methods, we can declare how our activity behaves when the user leaves and re-enters the activity. For example, if we're building a streaming video player, we might pause the video and terminate the network connection when the user switches to another app. When the user returns, we can reconnect to the network and allow the user to resume the video from the same spot. In other words, each callback allows us to perform specific work that's appropriate to a given change of state. Doing the right work at the right time and handling transitions properly make our app more robust and performant. For example, good implementation of the lifecycle callbacks can help ensure that our app avoids:

- Crashing if the user receives a phone call or switches to another app while using our app.
- Consuming valuable system resources when the user is not actively using it.
- Losing the user's progress if they leave our 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.


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

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.

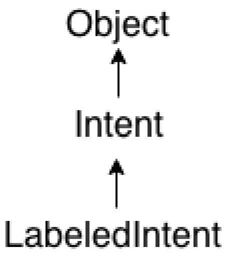
Declaring activities in the manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.stockquote"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="9"
        android:targetSdkVersion="18" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <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>

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

    </application>
```

```
<manifest ... >
  <application ... >
      <activity android:name=".ExampleActivity" />
      ...
  </application ... >
  ...
</manifest >
```

Connecting activities with intents

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. The **dictionary meaning** of intent is *intention or purpose*. So, it can be described as the intention to do action. The LabeledIntent is the subclass of android.content.Intent class.



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.

Types of Android Intents

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, we 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);
```

2) Explicit Intent

Explicit Intent specifies the component. In such case, intent provides the external class to be invoked.

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

Android Implicit Intent Example

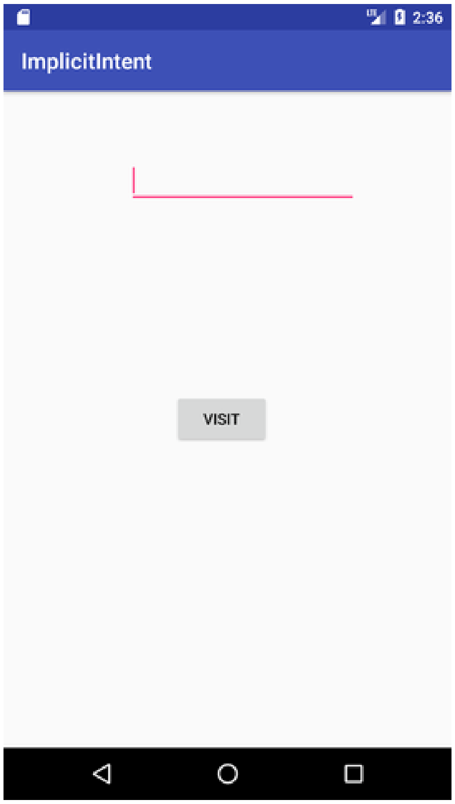
activity_main.xml

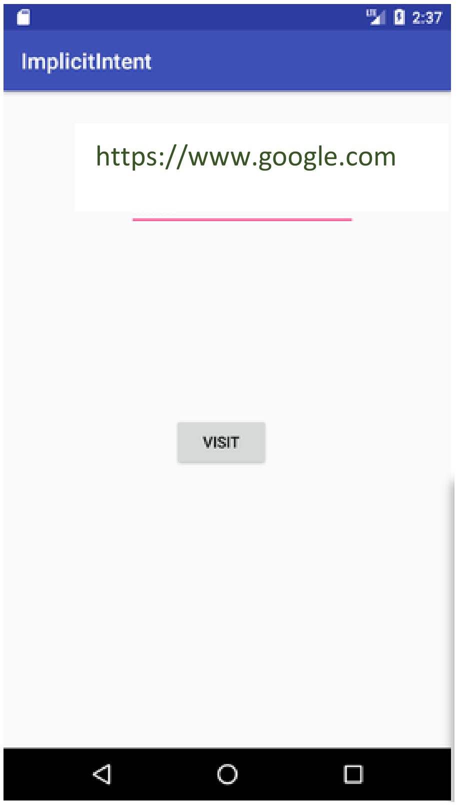
```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context=".MainActivity">
  <EditText
    android:id="@+id/editText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    />
  <Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Visit"
    app:layout_constraintTop_toBottomOf="@+id/editText"
 />
</android.support.constraint.ConstraintLayout>
```

MainActivity.java

```
import android.content.Intent;
import android.net.Uri;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {

    Button button;
    EditText editText;

    @Override
      protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        button = findViewById(R.id.button);
        editText =  findViewById(R.id.editText);

        button.setOnClickListener(new View.OnClickListener() {
            @Override
      public void onClick(View view) {
                String url=editText.getText().toString();
     Intent intent=new Intent(Intent.ACTION_VIEW, Uri.parse(url));
                startActivity(intent);
            }
        });
    }
}
```

Output





https://www.google.com

Android calling one activity from another activity example

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemasn
droid.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=".FirstActivity">

   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="First Activity"
      />

  <Button
      android:id="@+id/button"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:onClick="callSecondActivity"
      android:text="Call second activity"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```

ActivityOne class

File: MainActivityOne.java

```
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;

public class FirstActivity extends AppCompatActivity {


  @Override
  protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_first);
  }
  public void callSecondActivity(View view){
      Intent i = new Intent(getApplicationContext(), SecondActivity.class);
      i.putExtra("Value1", "Android By Javatpoint");
      i.putExtra("Value2", "Simple Tutorial");
      // Set the request code to any code you like, you can identify t
he
      // callback via this code
      startActivity(i);
  }
 }
```

activitytwo_main.xml

File: activitytwo_main.xml

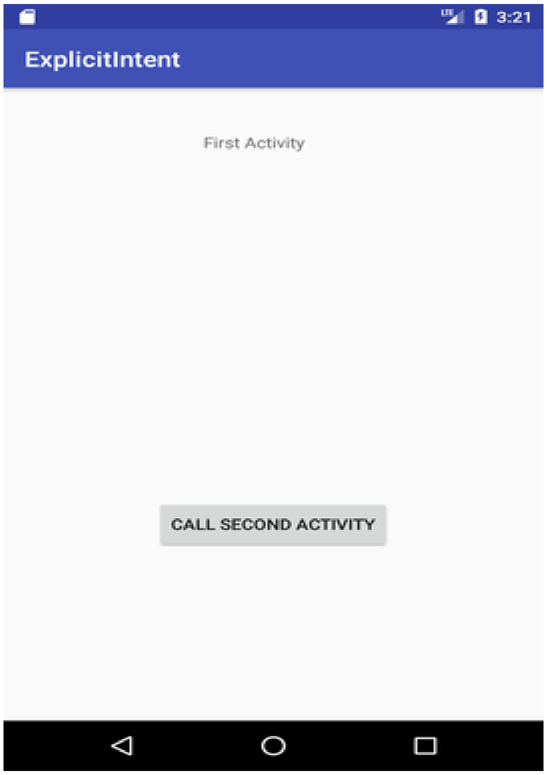
```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.
androd.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context=".SecondActivity">

   <TextView
      android:layout_width="wrap_content"
       android:layout_height="wrap_content”
       android:text="Second Activity"
/>

   <Button
      android:id="@+id/button"
       android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:onClick="callFirstActivity"
       android:text="Call first activity"  />
</android.support.constraint.ConstraintLayout>
```

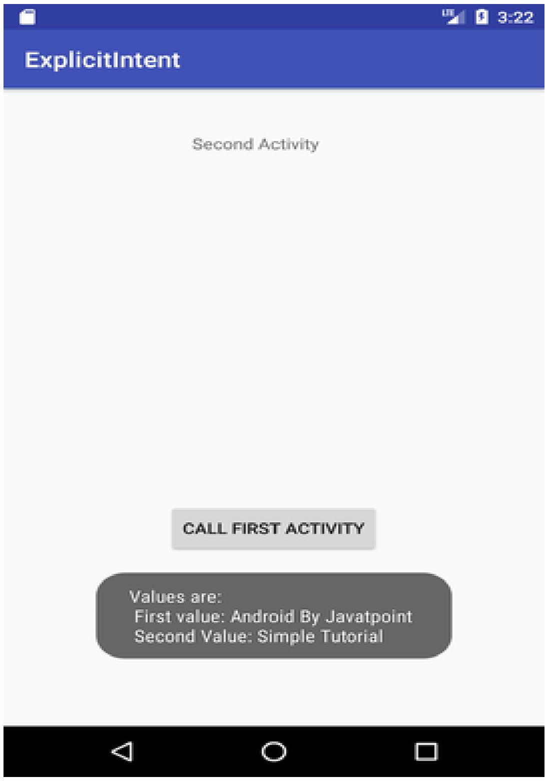
ActivityTwo class

File: MainActivityTwo.java

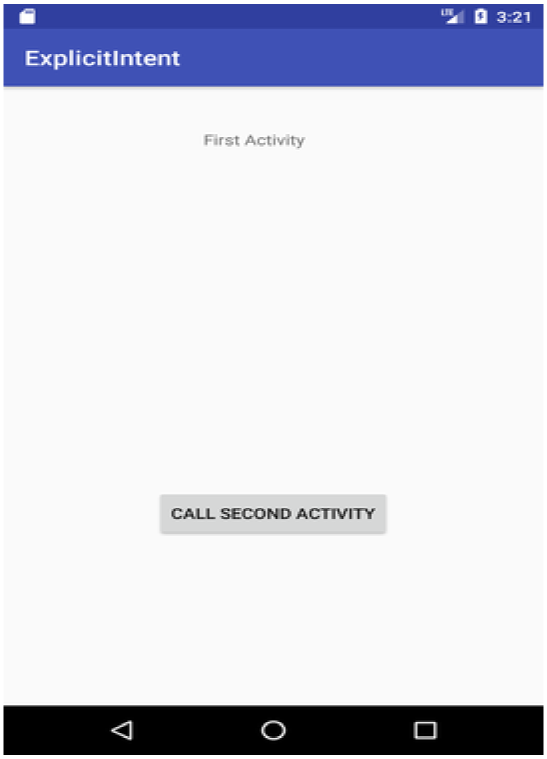
```
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Toast;

public class SecondActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);
        Bundle extras = getIntent().getExtras();
        String value1 = extras.getString("Value1");
        String value2 = extras.getString("Value2");
        Toast.makeText(getApplicationContext(),"Values are:\n First valu
e: "+value1+
                "\n Second Value: "+value2, Toast.LENGTH_LONG).sh
ow();
    }
    public void callFirstActivity(View view){
        Intent i = new Intent(getApplicationContext(), FirstActivity.class)
;
        startActivity(i);
    }
}
```

Output:





Values are:
First value: Android By Javatpoint
Second Value: Simple Tutorial



Passing data between activities

```
<resources>
<string name="app_name">Activity2</string>
<string name="message_txt">Enter Your message</string>
<string name="send_button_text">Send</string>
<string name="title_activity_second">SecondActivity</string>
</resources>
```

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"
>
<EditText
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:id="@+id/message_text"
android:hint="@string/message_txt"
/>
<Button
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:hint="@string/send_button_text"
android:onClick="sendmessage"
/>
</LinearLayout>
```

MainActivity.java

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

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

EditText message_text;

public final static String  MESSAGE_KEY ="com.example.message_key";

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

public void sendmessage (View view)
{
message_text = (EditText) findViewById(R.id.message_text);

String message = message_text.getText().toString();

Intent intent= new Intent(this ,SecondActivity.class);

intent.putExtra(MESSAGE_KEY,message);

startActivity(intent);
}

}
```

SecondActivity.java

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

public class SecondActivity extends Activity {

public final static String MESSAGE_KEY ="com.example.message_key";

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
Intent intent = getIntent();
String message = intent.getStringExtra(MESSAGE_KEY);
TextView textView = new TextView(this);
textView.setTextSize(45);
textView.setText(message);
setContentView(textView);

}

}
```

activity_second.xml.

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
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=".SecondActivity">

<android.support.design.widget.AppBarLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:theme="@style/AppTheme.AppBarOverlay">

<android.support.v7.widget.Toolbar
android:id="@+id/toolbar"
android:layout_width="match_parent"
android:layout_height="?attr/actionBarSize"
android:background="?attr/colorPrimary"
app:popupTheme="@style/AppTheme.PopupOverlay" />

</android.support.design.widget.AppBarLayout>

<include layout="@layout/content_second" />

</android.support.design.widget.CoordinatorLayout>
```

Getting a result back from a child activity

MainActivity.Java

```
package com.example.simpleinterestcalculator;

import androidx.*;

public class MainActivity extends AppCompatActivity {

    private EditText ptxt, rtxt, ttxt;
    private Button sbtn;
    private TextView resulttxt;

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

        ptxt = findViewById(R.id.p);
        rtxt = findViewById(R.id.r);
        ttxt = findViewById(R.id.t);
        sbtn = findViewById(R.id.sendbtn);
        resulttxt = findViewById(R.id.result);

        sbtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (ptxt.getText().toString().equals("") ||
rtxt.getText().toString().equals("") || ttxt.getText().toString().equals("")) {
                    Toast.makeText(MainActivity.this, "Insert all Information",
Toast.LENGTH_LONG).show();
                } else {
                    int p = Integer.parseInt(ptxt.getText().toString());
                    int t = Integer.parseInt(rtxt.getText().toString());
                    int r = Integer.parseInt(ttxt.getText().toString());

                    Intent i = new Intent(MainActivity.this, SecondActivity.class);
                    i.putExtra("p", p);
                    i.putExtra("t", t);
                    i.putExtra("r", r);
                    startActivityForResult(i, 1);

                }          }
        });

    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if(requestCode==1 && resultCode==2)
        {
            int si = data.getIntExtra("result",0)      ;
             resulttxt.setText("Simple Interest : " + si);
        }
    }
}//end of class
```

SecondActivity.java

```java
package com.example.simpleinterestcalculator;

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 SecondActivity extends AppCompatActivity {


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);
        setTitle("Simple Interest Calculator");
        Intent in = getIntent();
        int p = in.getIntExtra("p",0);
        int t = in.getIntExtra("t",0);
        int r = in.getIntExtra("r",0);

        Button calsibtn = findViewById(R.id.sibtn);
        TextView inputtxt = findViewById(R.id.inputtxt);

        inputtxt.setText("Principal :" + p +"\n"+ "Time : "+t +"\n" + " Rate :
"+r);

        calsibtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent();
                int result = (p*t*r)/100;
                i.putExtra("result" , result);
                setResult(2 , i);
                finish();

            }
        });


    }
}
```

Getting and setting data to/from the layout file.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:background="#FFFF8D"
   tools:context=".MainActivity">

   <TextView
      android:id="@+id/textView"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignParentLeft="true"
      android:layout_alignParentStart="true"
      android:layout_alignParentTop="true"
      android:text="NAME"
      android:textSize="20sp"
      android:layout_margin="20dp" />

   <TextView
      android:id="@+id/textView2"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:textSize="20sp"
      android:text="PASSWORD"
      android:layout_marginTop="38dp"
```

```
    android:layout_below="@+id/textView"

    android:layout_alignLeft="@+id/textView"

    android:layout_alignStart="@+id/textView" />


<EditText

    android:id="@+id/editName"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:ems="10"

    android:inputType="textPersonName"

    android:hint="Enter Name"

    android:layout_alignParentTop="true"

    android:layout_alignParentRight="true"

    android:layout_alignParentEnd="true"

    android:layout_alignLeft="@+id/editPassword"

    android:layout_alignStart="@+id/editPassword" />


<EditText

    android:id="@+id/editPassword"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:ems="10"

    android:hint="Enter Password"

    android:inputType="textPassword"

    android:layout_alignBottom="@+id/textView2"

    android:layout_alignParentRight="true"

    android:layout_alignParentEnd="true"

    android:layout_marginRight="18dp"

    android:layout_marginEnd="18dp" />
    <Button
        android:id="@+id/buttonSubmit"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_below="@+id/textView2"
        android:layout_marginTop="20dp"
        android:text="SUBMIT" />
    <Button
        android:id="@+id/buttonReset"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="RESET"
        android:layout_alignBaseline="@+id/buttonSubmit"
        android:layout_alignBottom="@+id/buttonSubmit"
        android:layout_centerHorizontal="true" />
    <TextView
        android:id="@+id/tvResult"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginBottom="143dp"
        android:textSize="30sp" />
</RelativeLayout>
```

MainActivity.java

```
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
   // These are the global variables
   EditText editName, editPassword;
   TextView result;
   Button buttonSubmit, buttonReset;

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

      editName  = (EditText) findViewById(R.id.editName);
      editPassword = (EditText) findViewById(R.id.editPassword);
      result = (TextView) findViewById(R.id.tvResult);
      buttonSubmit = (Button) findViewById(R.id.buttonSubmit);
      buttonReset = (Button) findViewById(R.id.buttonReset);
      /*
         Submit Button
      */
      buttonSubmit.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
             String name = editName.getText().toString();
             String password = editPassword.getText().toString();
             result.setText("Name:\t" + name + "\nPassword:\t" + password );

         }
      });

      /*
         Reset Button
      */
      buttonReset.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
             editName.setText("");
             editPassword.setText("");
             result.setText("");
             editName.requestFocus();
         }
      });
   }
}
```