# Unit 1 – Lesson 4. Create a Simple Calculator App (Android Activity and Events –Part 2)

**Aim:**

1. How do we set up the UI view through the XML file in Android?
2. How do we use XML file layouts?
3. How do we use call and return variables on the View in our Java class?
4. How do we set up our java class and what method(s) shall we use?

**Objectives:** After the lesson, students will be able to:

1. Obtain basic understanding of Android app layout
2. Obtain basic understanding of Android TextView
3. Understand the importance of android:id
4. Set up the main activity class and onClick(View ) method

**CLASS PROCEDURE:**

***Do Now:***

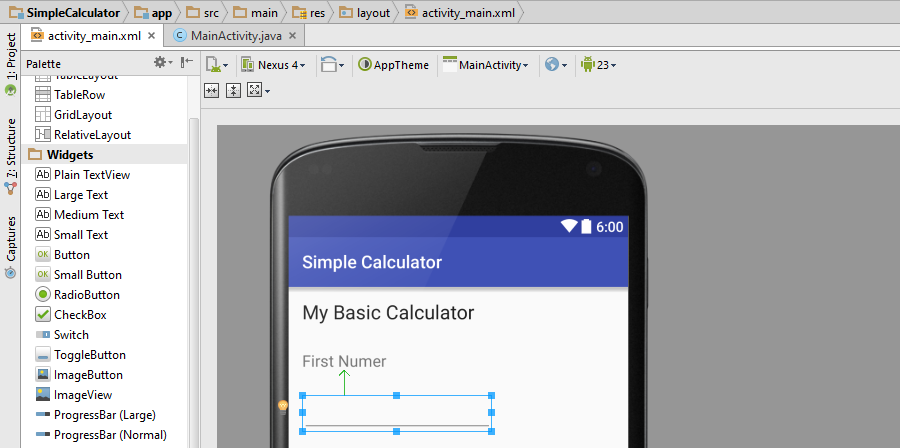
Open Android Studio. Create a new empty Android project and name it “Simple Calculator”.

1. Verify that your project location is pointed to the USB flash drive,
2. Make sure that your “Activity Name” is MainActivity (by default), and your “Layout Name” is Activity\_Main (by default).
3. After the Android Studio finishes creating the project, go to the top menu bar 🡪 Window 🡪 choose “Restore Default Layout” or press Shift + F12

***Class Activity / Lab:***

Follow the following steps to create a simple calculator:

First, we need to set up the UI layout:



1. Click “Activity\_Main.xml”, and then choose “Design” view
2. Remove “Hello World” from the screen
3. From the “Palette” on the left, drag “Large Text” and drop it on your design. Change the “Text” to “My Basic Calculator”, leave the ID as “textView”, align the text on the top, and choose a color (i.e., black) for the text. You may adjust the appearance through the “Properties” panel.
4. Next, drag and drop a “Medium Text” to the design, change the text to “First Number”, ID to “textView2”. You may adjust the appearance through the “Properties” panel.
5. Then, from the “Palette” on the left, navigate down and find “Text Field”, drag “Number” and drop to your design, change the ID for the Text Field to “etFirstNumber”. You may adjust the appearance through the “Properties” panel.
6. Repeat the above steps and add the “Second Number” text and the second number text field.
7. On the Palette 🡪 Widgets, drag and drop four buttons to the design. Add buttons of “+”, “-”, “x”, “/” operations. Make sure the IDs for the buttons are named btnAdd, btnSubtract, btnMultiply, and btnDivide.
8. Add Text “Result” and TextView for result to the design. Assign tvResult to the ID for your TextView.

Second, we need to set up the Java program for the MainActivity class:

1. Navigate to the top menu and find “src” 🡪 “main” 🡪 “java”, and choose “MainActivity.java”.
2. First, we need to verify that all of the packages have been imported correctly:

import android.support.v7.app.ActionBarActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

1. Android Studio creates the class header for you and our MainActivity class should extends the ActionBarActivity class and implements the View.onClickListener interface

public class MainActivity extends ActionBarActivity implements View.OnClickListener

1. Declare fields (don’t forget the encapsulation rule in Java):

    private Button btnAdd,btnsub,btndivide,btnmul;

    private TextView tvresult;

    private EditText etfirst,etsecond;

1. Create a void method init() to instantiate the fields (buttons, EditText, TextView, etc.)

    private void init() {

        btnAdd =(Button)findViewById(R.id.btnAdd);

        btnsub = (Button)findViewById(R.id.btnSubtract);

        btndivide = (Button)findViewById(R.id.btnDivide);

        btnmul = (Button)findViewById(R.id.btnMultiply);

        etfirst = (EditText)findViewById(R.id.etFirstNumber);

        etsecond =(EditText)findViewById(R.id.etSecondNumber);

        tvresult = (TextView)findViewById(R.id.tvResult);

    }

1. Add the ini() to the void onCreate() method that has already been auto generated by Android Studio.
2. Now create the onClick() method to define the event after the buttons being clicked on:

    public void onClick(View view) {

        String num1 = etfirst.getText().toString();

        String num2 = etsecond.getText().toString();

        switch(view.getId()){

            case R.id.btnAdd:

                int addition = Integer.parseInt(num1) + Integer.parseInt(num2);

                tvresult.setText(String.valueOf(addition));

                break;

            case R.id.btnSubtract:

                int subtraction = Integer.parseInt(num1) - Integer.parseInt(num2);

                tvresult.setText(String.valueOf(subtraction));

                break;

            case R.id.btnDivide:

                try{

                    int division = Integer.parseInt(num1) / Integer.parseInt(num2);

                    tvresult.setText(String.valueOf(division));

                }catch(Exception e){

                    tvresult.setText("Cannot DIVIDE!");

                }

                break;

            case R.id.btnMultiply:

                int multiply = Integer.parseInt(num1) \* Integer.parseInt(num2);

                tvresult.setText(String.valueOf(multiply));

                break;

        }

1. Add the onClick() to init() method:

        btnAdd.setOnClickListener(this);

        btnsub.setOnClickListener(this);

        btndivide.setOnClickListener(this);

        btnmul.setOnClickListener(this);

1. Save, test and run the project on the emulator.

***Discussion / Presentation:***

1. In the ini() method, we called R.id. What is the role of R (or R file)?

* Android generates a R file and the R file contains IDs for all the resources in the res folder of your project and also some additional IDs that you define on your own (in the layouts, for example).
* The IDs are needed for the Android resource management system to retrieve the files from the APK. Each ID is basically a number which corresponds to some resource in the resource management system.

1. What would happen if the ID in XML and the ID in Java methods do not match?
2. How do we interpret some of the common error messages?

***Solutions / Source Code for the lab:***

Source code for Activity\_Main.xml:

<RelativeLayout xmlns:android="<http://schemas.android.com/apk/res/android>"

    xmlns:tools="<http://schemas.android.com/tools>" android:layout\_width="match\_parent"

    android:layout\_height="match\_parent" android:paddingLeft="5dip"

    android:paddingRight="5dip"

    android:paddingTop="5dip"

    android:paddingBottom="5dip"

tools:context=".MainActivity">

    <TextView

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:textAppearance="?android:attr/textAppearanceLarge"

        android:text="My Basic Calculator"

        android:id="@+id/textView"

        android:layout\_alignParentTop="true"

        android:layout\_centerHorizontal="true"

        android:textColor="#008888" />

    <TextView

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:textAppearance="?android:attr/textAppearanceMedium"

        android:text="First Number"

        android:id="@+id/textView2"

        android:layout\_below="@+id/textView"

        android:layout\_alignParentLeft="true"

        android:layout\_alignParentStart="true"

        android:layout\_marginTop="30dp" />

    <EditText

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:inputType="number"

        android:ems="10"

        android:id="@+id/etFirstNumber"

        android:layout\_below="@+id/textView2"

        android:layout\_alignParentLeft="true"

        android:layout\_alignParentStart="true" />

    <TextView

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:textAppearance="?android:attr/textAppearanceMedium"

        android:text="Second Number"

        android:id="@+id/textView3"

        android:layout\_below="@+id/etFirstNumber"

        android:layout\_alignParentLeft="true"

        android:layout\_alignParentStart="true" />

    <EditText

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:inputType="number"

        android:ems="10"

        android:id="@+id/etSecondNumber"

        android:layout\_below="@+id/textView3"

        android:layout\_alignParentLeft="true"

        android:layout\_alignParentStart="true" />

    <LinearLayout

        android:orientation="horizontal"

        android:layout\_width="fill\_parent"

        android:layout\_height="wrap\_content"

        android:layout\_below="@+id/etSecondNumber"

        android:layout\_alignParentLeft="true"

        android:layout\_alignParentStart="true">

        <Button

            android:layout\_width="wrap\_content"

            android:layout\_height="wrap\_content"

            android:text="+"

            android:id="@+id/btnAdd"

            android:background="#51ff00" />

        <Button

            android:layout\_width="wrap\_content"

            android:layout\_height="wrap\_content"

            android:text="-"

            android:id="@+id/btnSubtract"

            android:background="#ff0000" />

        <Button

            android:layout\_width="wrap\_content"

            android:layout\_height="wrap\_content"

            android:text="/"

            android:id="@+id/btnDivide"

            android:background="#ffb300" />

        <Button

            android:layout\_width="wrap\_content"

            android:layout\_height="wrap\_content"

            android:text="\*"

            android:id="@+id/btnMultiply"

            android:background="#00ffe6" />

    </LinearLayout>

    <LinearLayout

        android:orientation="horizontal"

        android:layout\_width="fill\_parent"

        android:layout\_height="wrap\_content"

        android:layout\_centerVertical="true"

        android:layout\_alignParentRight="true"

        android:layout\_alignParentEnd="true">

        <TextView

            android:layout\_width="wrap\_content"

            android:layout\_height="wrap\_content"

            android:textAppearance="?android:attr/textAppearanceLarge"

            android:id="@+id/tvResult"

            android:layout\_centerVertical="true"

            android:layout\_alignParentLeft="true"

            android:layout\_alignParentStart="true"

            android:textColor="#0000FF"

            android:layout\_marginTop="40dp" />

    </LinearLayout>

</RelativeLayout>

Source code for MainActivity.java:

package com.basiccalc.mybasiccalculator;

import android.support.v7.app.ActionBarActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

public class MainActivity extends ActionBarActivity implements View.OnClickListener {

    private Button btnAdd,btnsub,btndivide,btnmul;

    private TextView tvresult;

    private EditText etfirst,etsecond;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_main);

        init();

    }

    private void init() {

        btnAdd =(Button)findViewById(R.id.btnAdd);

        btnsub = (Button)findViewById(R.id.btnSubtract);

        btndivide = (Button)findViewById(R.id.btnDivide);

        btnmul = (Button)findViewById(R.id.btnMultiply);

        etfirst = (EditText)findViewById(R.id.etFirstNumber);

        etsecond =(EditText)findViewById(R.id.etSecondNumber);

        tvresult = (TextView)findViewById(R.id.tvResult);

        btnAdd.setOnClickListener(this);

        btnsub.setOnClickListener(this);

        btndivide.setOnClickListener(this);

        btnmul.setOnClickListener(this);

    }

    @Override

    public void onClick(View view) {

        String num1 = etfirst.getText().toString();

        String num2 = etsecond.getText().toString();

        switch(view.getId()){

            case R.id.btnAdd:

                int addition = Integer.parseInt(num1) + Integer.parseInt(num2);

                tvresult.setText(String.valueOf(addition));

                break;

            case R.id.btnSubtract:

                int subtraction = Integer.parseInt(num1) - Integer.parseInt(num2);

                tvresult.setText(String.valueOf(subtraction));

                break;

            case R.id.btnDivide:

                try{

                    int division = Integer.parseInt(num1) / Integer.parseInt(num2);

                    tvresult.setText(String.valueOf(division));

                }catch(Exception e){

                    tvresult.setText("Cannot DIVIDE!");

                }

                break;

            case R.id.btnMultiply:

                int multiply = Integer.parseInt(num1) \* Integer.parseInt(num2);

                tvresult.setText(String.valueOf(multiply));

                break;

        }

    }

}