Create Android Sample Application:

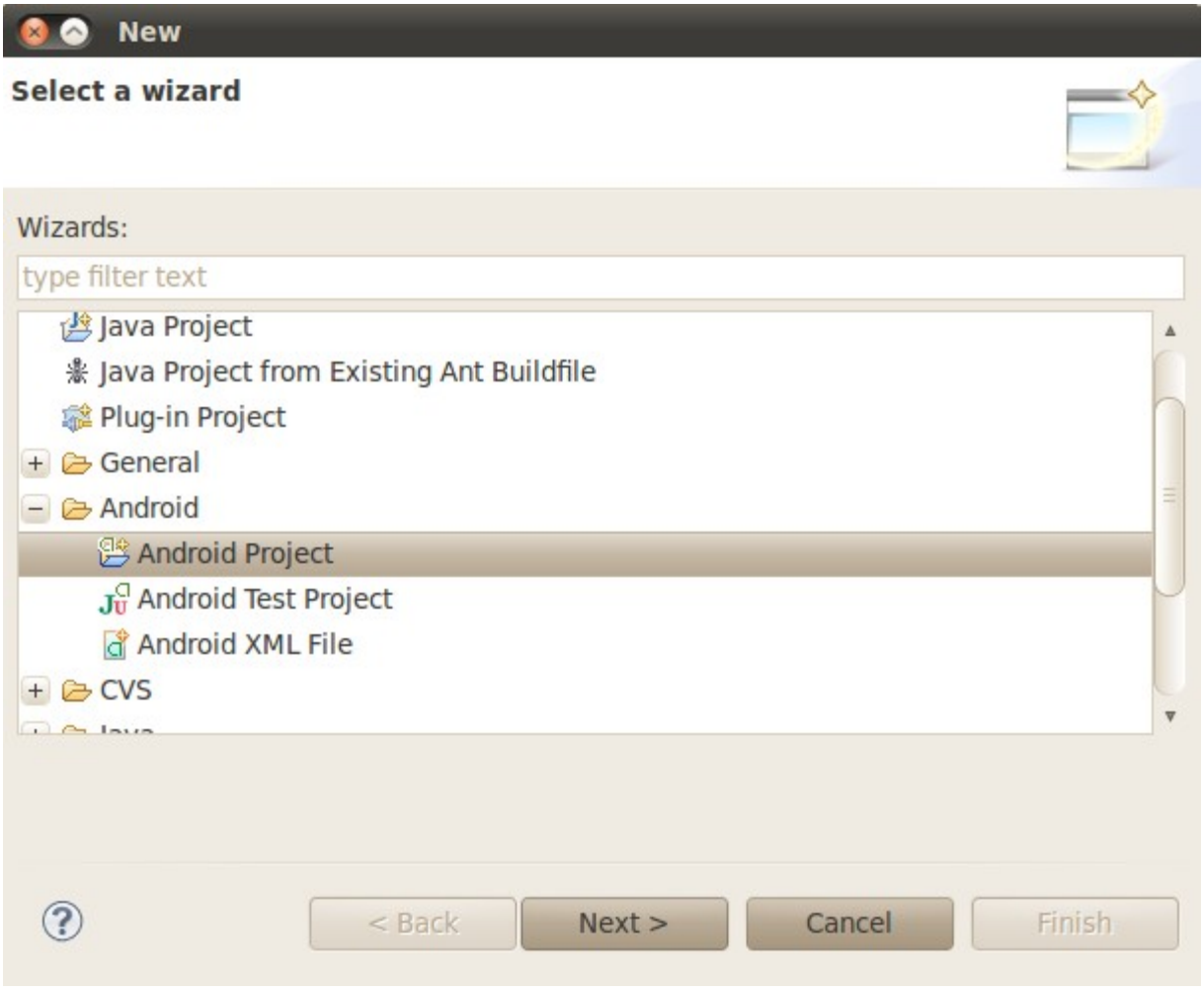
After setting up working environment we will start designing our own sample application, which we will test using *Robotium* in next section.

Our sample application would be a simple calculator to multiply two integer/decimal values. It will take two inputs and on clicking 'Multiply' it will show their multiply result.

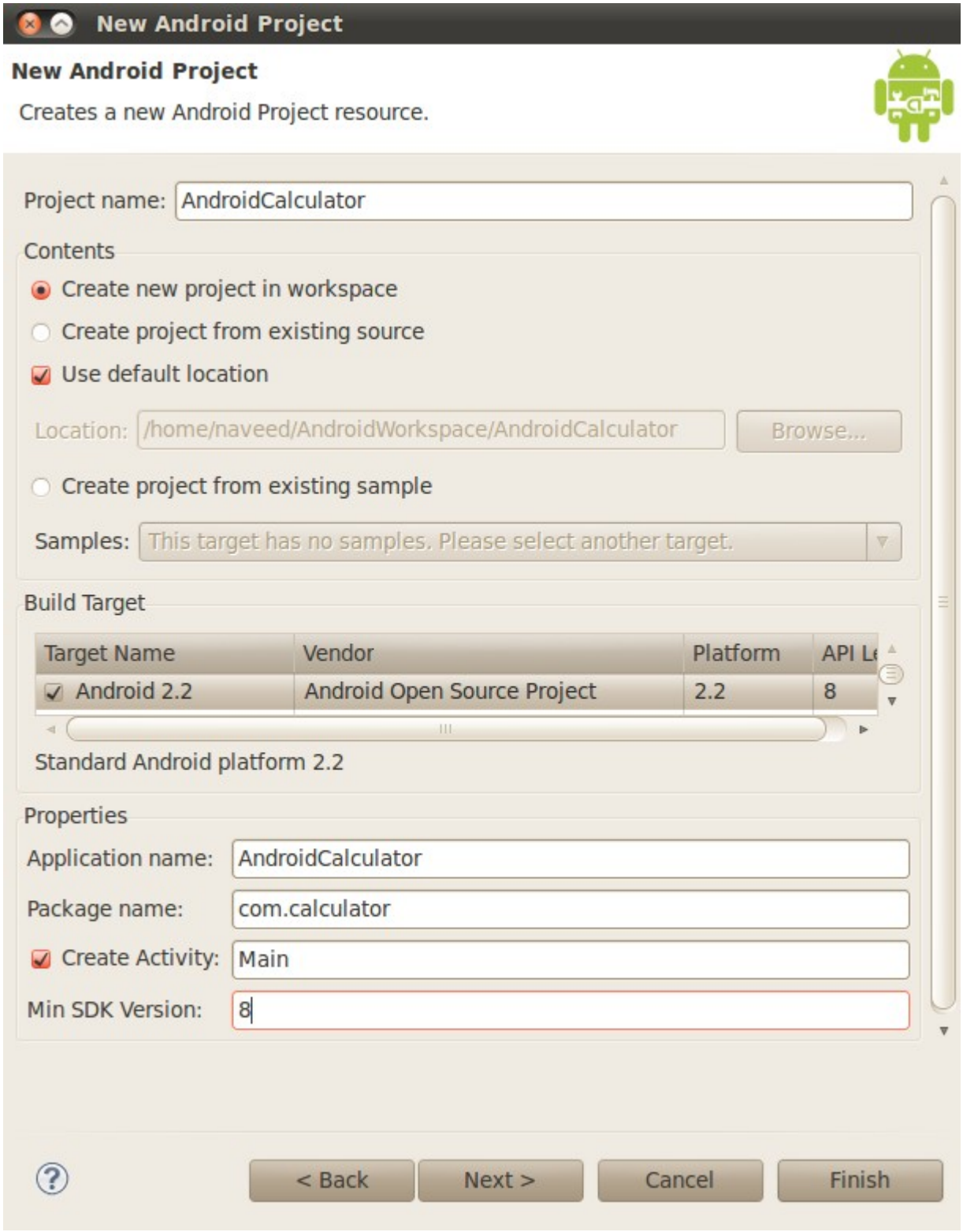
For simplicity steps are categorized as,

1. Create Project

Click on *File menu*, select *New* and click on the *Others*,
From *New* window, Drag down to *Android* option, expand it, and select *Android Project* and Click on *Next.*



From *New Android Project Window*, enter *Project Name* as 'AndroidCalculator', From *Contents* section select 'Create New Project in workspace', Check Android 2.2 from *Build Target* section, *Application Name* 'AndroidCalculator', *Package Name* 'com.calculator', *Create Activity* 'Main' and '8' as *Min SDK Version*

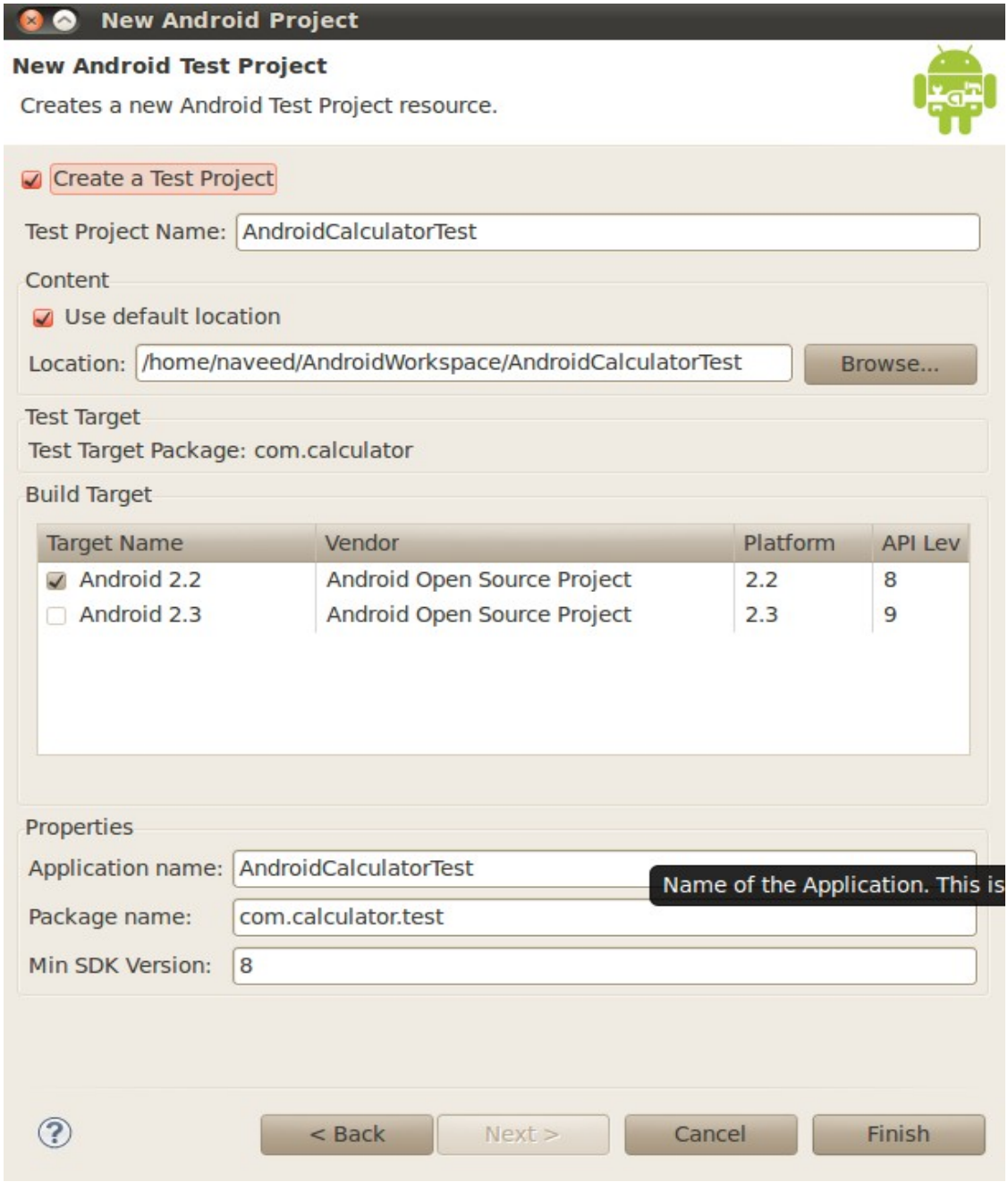


Note: you can enter any other options best suits to your need.

Click on Next and it will load *New Android Project window eclipse offer you to create project to test your project*, at this time we will avail it to test 'AndroidCalculator' we made in last step. We can skip this part by un-checking *Create a Test Project* option, if you want to manually create test project later, we will create test project right away.

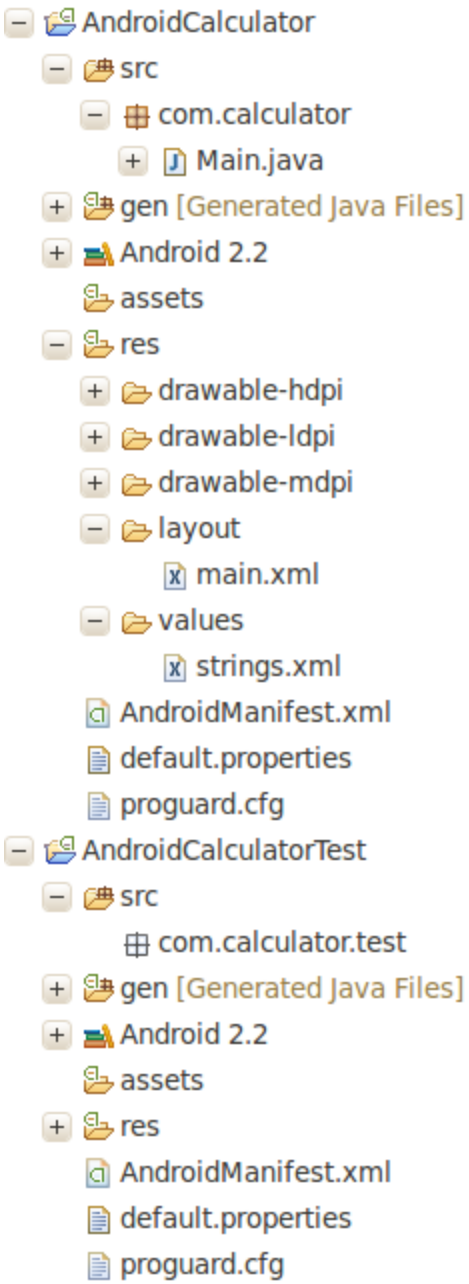
2. Create Test Project

Check *Create a Test Project* and it will automatically fill rest of fields based on the last made project (AndroidCalculator), and click on Finish.



Note: As our current focus is to design sample project so for now, we will just create test project and in next section we will be working on this new created project to test AndroidCalculator.

After successfully creating projects, our Project explorer screen should look like,



Now two projects are created, we will simply work on first project to design our sample calculator application.

3. Understanding Project Architect

Expand the *src* directory then expand *com.calculator* directory, *Main.java* file contains application logic.

In *rec* directory we can define application's UI interface.
In *Main.xml* we can put controls on application interface and in *string.xml* we can define their string values,which would be visible on UI.

We would not get into details, as its not in our scope so far.

4. Designing Layout

In main.xml enter following code and save it.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:orientation="vertical"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
   >
<TextView
   android:layout_width="fill_parent"
   android:layout_height="wrap_content"
   android:text="@string/hello"
   />

<TextView
          android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="@string/txtSpace"
/>

<TextView
          android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="@string/txtFirstNumber"
/>

<EditText
          android:inputType="numberDecimal"
          android:id="@+id/EditText01"
          android:layout_width="fill_parent"
          android:layout_height="wrap_content">
</EditText>

<TextView
          android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="@string/txtSpace"
/>

<TextView
          android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="@string/txtSecondNumber"
/>

<EditText
          android:inputType="numberDecimal"
          android:id="@+id/EditText02"
          android:layout_width="fill_parent"
          android:layout_height="wrap_content">
</EditText>

<TextView
          android:layout_width="wrap_content"
   android:layout_height="wrap_content"
```

```
   android:text="@string/txtSpace"
/>

<TextView
        android:id="@+id/TextView01"
        android:layout_width="wrap_content"
   android:layout_height="wrap_content"
/>

<TextView
        android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="@string/txtSpace"
/>

<Button
        android:text="Multiply"
        android:id="@+id/Button01"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content">
</Button>

</LinearLayout>
```

In *string.xml* enter following code

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

<resources>

   <string name="hello">Enter two values and click on Calculate to multiply them.</string>
   <string name="app_name">AndroidSimpleCalculator</string>
   <string name="txtFirstNumber">Enter First Number</string>
   <string name="txtSecondNumber">Enter Second Number</string>
   <string name="txtSpace"></string>
</resources>
```

5. Designing Application Logic

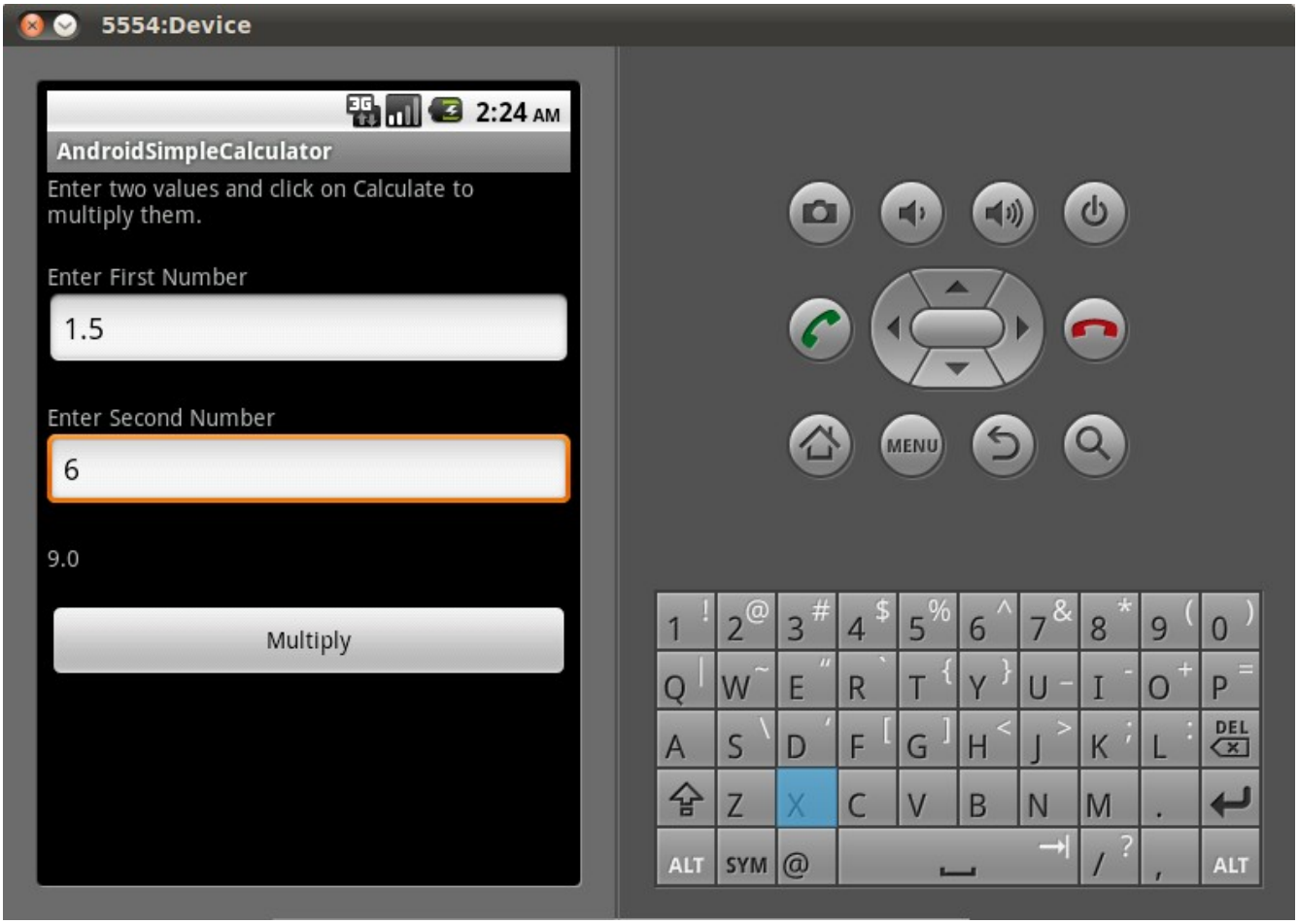
In Main.java enter following code and save it.

```
package com.calculator;

import com.calculator.R;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.text.Editable;

public class Main extends Activity {
```

```
        EditText FirstValue;
        EditText SecondValue;
        TextView Result;
        Button Calculate;
        float num1 , num2;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        FirstValue = (EditText) findViewById(R.id.EditText01);

        SecondValue = (EditText) findViewById(R.id.EditText02);

        Result = (TextView) findViewById(R.id.TextView01);
        Result.setText("0.00");

        Calculate = (Button) findViewById(R.id.Button01);

        //Adding listener to button
        Calculate.setOnClickListener(new View.OnClickListener() {

            public void onClick(View v) {
                    //Getting first & second values and passing to show result
                    showResult(FirstValue.getText(), SecondValue.getText());
            }
            });
}

//Showing multiply results
protected void showResult(Editable first, Editable second)
{
        float num1 = Float.parseFloat(first.toString());
        float num2 = Float.parseFloat(second.toString());
        float result = num1 * num2;
        Result.setText(String.valueOf(result));
}

}
```

Our application is designed and its time to run it.
Right click on project select Run As and then select Android Application & and wait for while.
It will load Android simulator, you need to wait for some time, it will launch application itself.

Our simple multiply calculator is ready, enter some integer/decimal values and click on Multiply, it will show the result above Click button.



Next: In next step we will work on AndroidCaculatorTest project to test that simple calculator.