Mobile App Development Lab Manual

PROGRAM 1

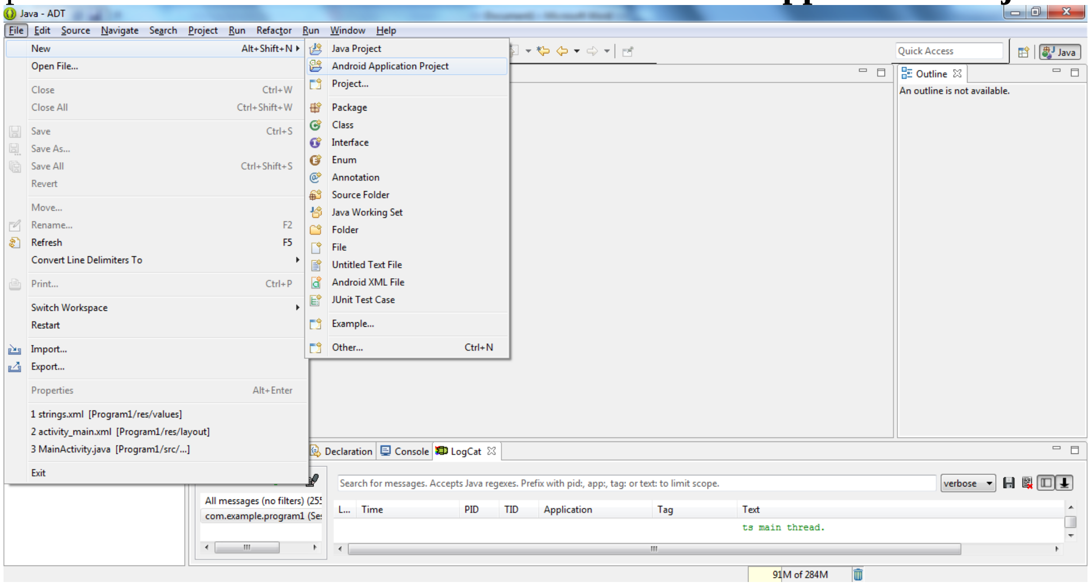
Aim:

To develop a Simple Android Application that uses GUI components, Font and Colors.

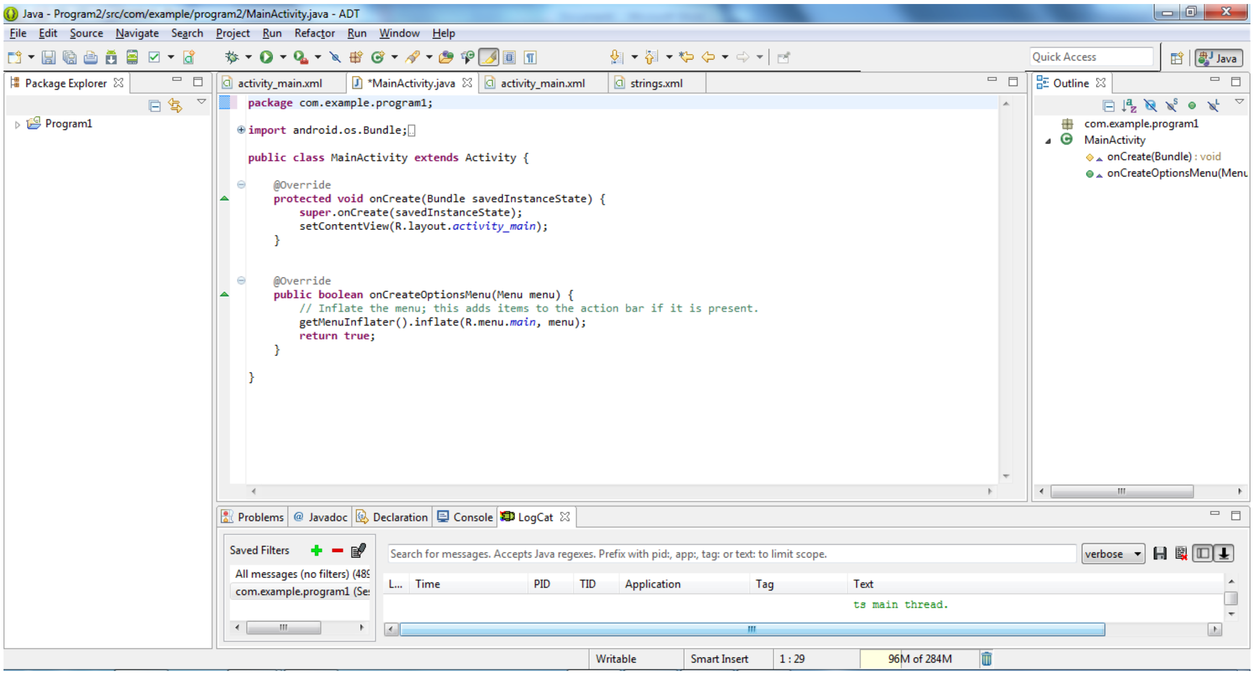
Procedure:

Creating a New project:

- Open IDE and then click on **File -> New -> Android Application Project.**



- Then type the Application name as “**Program1″** and click **Next.**
- Then click **Next.**
- Then click **Next.**
- Then select **Blank Activity** and click **Next.**
- Finally click F**inish**.
- It will take some time to build and to load the project.
- After completion it will look as given below.



```java
package com.example.program1;

import android.os.Bundle;

public class MainActivity extends Activity {

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


    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

}
```

```
ts main thread.
```

Designing layout for the Android Application:

- Click on **Program1 -> res -> layout -> activity_main.xml.**
- Now go to XML code editor.
- Then delete the code which is there and type the code as given below.

Code for Activity_main.xml:

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

    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="30dp"
        android:gravity="center"
        android:text="@string/hello_world"
        android:textSize="25sp"
        android:textStyle="bold" />

    <Button
        android:id="@+id/button1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="20dp"
        android:gravity="center"
        android:text="@string/font_size"
        android:textSize="25sp" />


<Button
        android:id="@+id/button2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="20dp"
        android:gravity="center"
        android:text="@string/color_size"
        android:textSize="25sp" />

</LinearLayout>
```

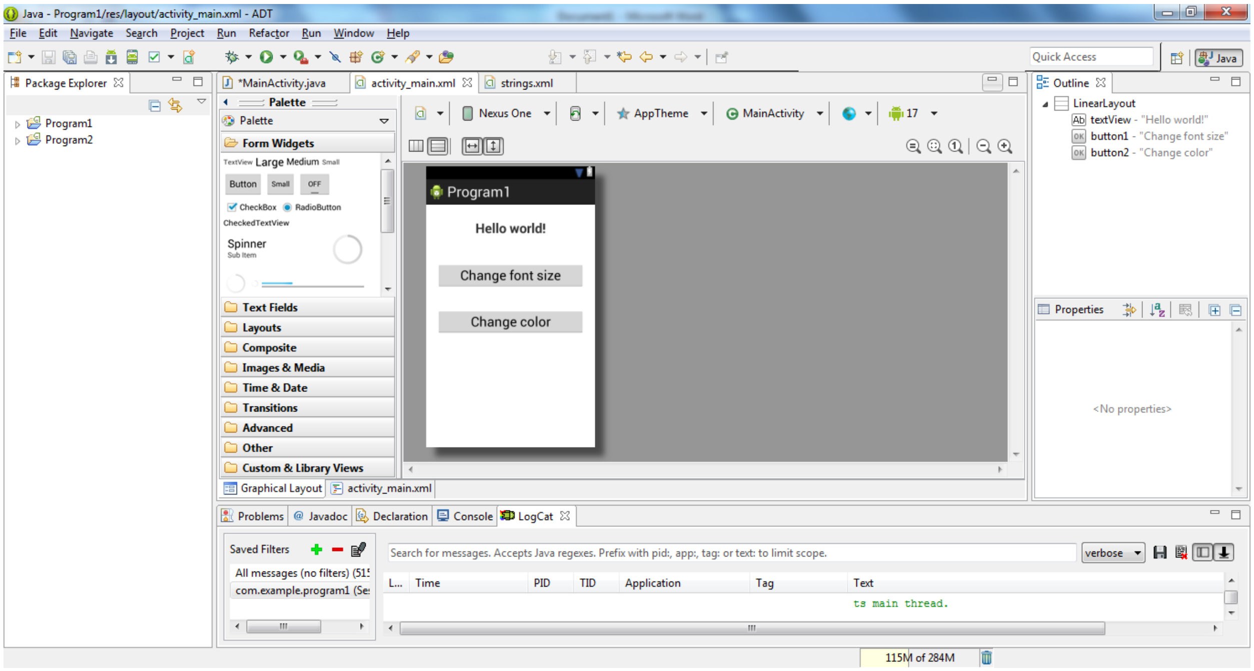
Defining strings for the Android Application:

- Click on **Program1 -> res -> values -> strings.xml.**
- Now go to XML code editor.
- Then delete the code which is there and type the code as given below.

Code for strings.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
<string name="app_name">Program1</string>
<string name="action_settings">Settings</string>
<string name="hello_world">Hello world!</string>
<string name="font_size">Change font size</string>
<string name="color_size">Change color</string>
</resources>
```

- Now click on Design of **Activity_main.xml** and your application will look as given below.



```
ts main thread.
```

- So now the designing and defining part is completed.

Java Coding for the Android Application:

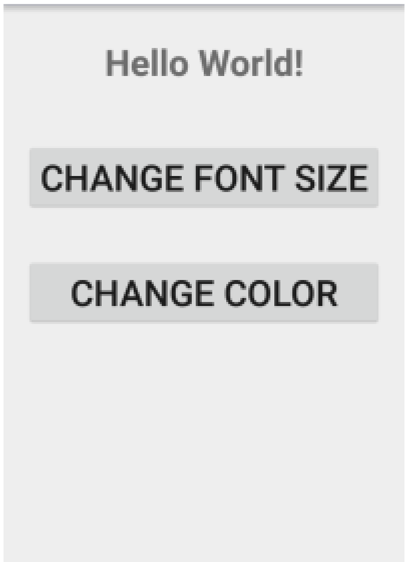
- Click on **Program1 -> src -> com.example.program1 -> MainActivity.**
- Then delete the code which is there and type the code as given below.

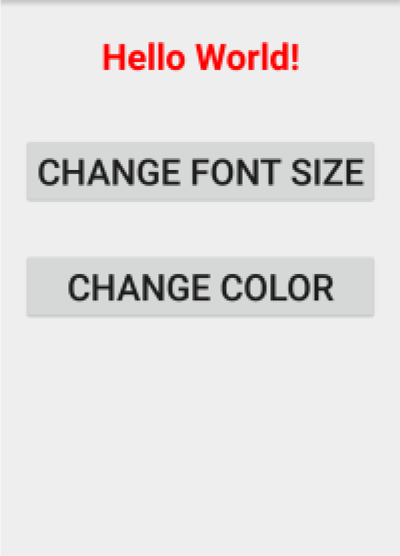
Code for MainActivity.java:

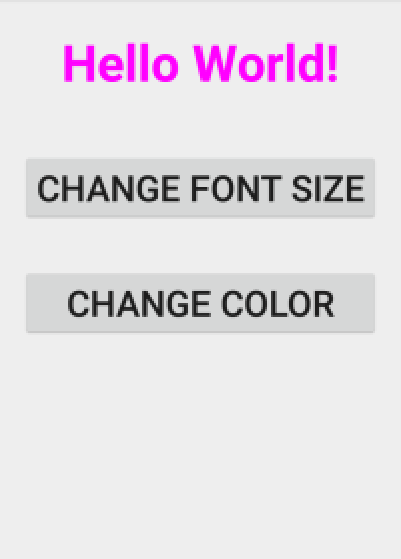
```
package com.example.program1;

import android.os.Bundle;
import android.app.Activity;
import android.graphics.Color;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class MainActivity extends Activity {

    //Code Start
    int ch=1;
    float font=30;
    //Code End

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

        //Code Start
        final TextView t= (TextView) findViewById(R.id.textView);
        Button b1= (Button) findViewById(R.id.button1);
        b1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                t.setTextSize(font);
                font = font + 5;
                if (font == 50)
                    font = 30;
            }
        });
        Button b2= (Button) findViewById(R.id.button2);
        b2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                switch (ch) {
                    case 1:
                        t.setTextColor(Color.RED);
                        break;
                    case 2:
                        t.setTextColor(Color.GREEN);
                        break;
                    case 3:
                        t.setTextColor(Color.BLUE);
                        break;
                    case 4:
                        t.setTextColor(Color.CYAN);
                        break;
                    case 5:
                        t.setTextColor(Color.YELLOW);
                        break;
                    case 6:
                        t.setTextColor(Color.MAGENTA);
                        break;
                }
                ch++;
                if (ch == 7)
                    ch = 1;
            }
        });
      //Code End
    }

}
```

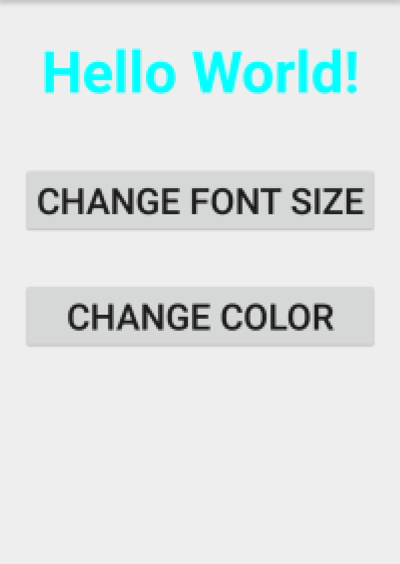
- So now the Coding part is also completed.
- Now run the application to see the output.

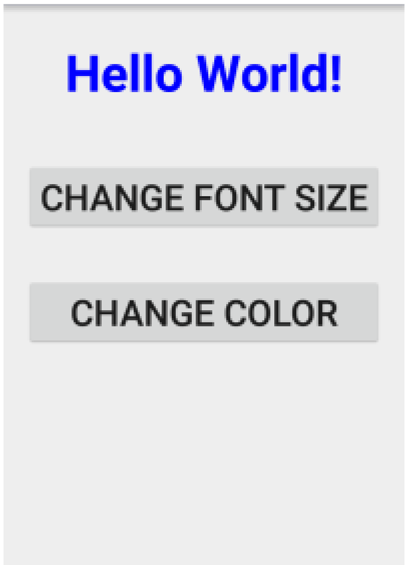
Output:

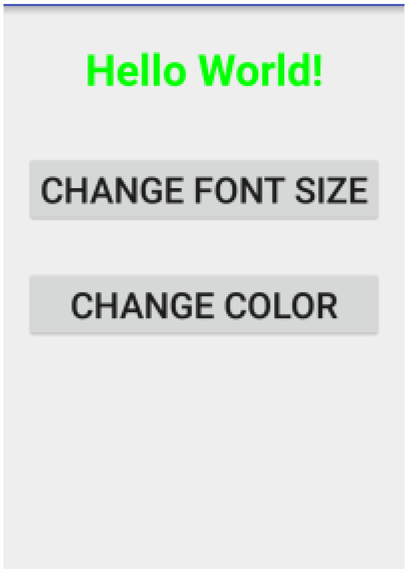












Result:

Thus a Simple Android Application that uses GUI components, Font and Colors is developed and executed successfully.

PROGRAM 2

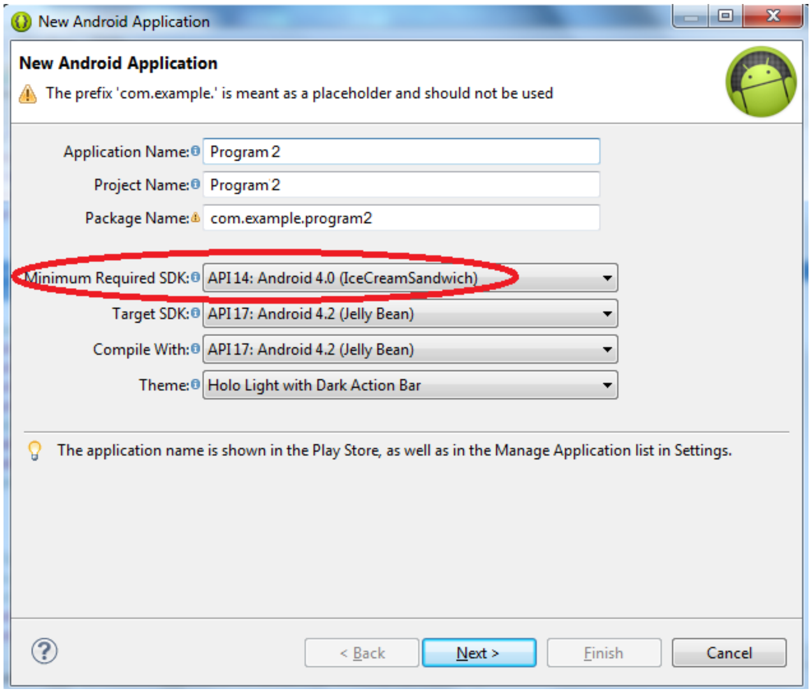
Aim:

To develop a Simple Android Application that uses Layout Managers and Event Listeners.

Procedure:

Creating a New project:

- Open IDE and then click on **File -> New -> Android Application Project.**
- Then type the Application name as “**Program2″, Minimim Required SDK = API 14** and click **Next.**



New Android Application

New Android Application

⚠ The prefix 'com.example.' is meant as a placeholder and should not be used

Application Name: Program 2

Project Name: Program 2

Package Name: com.example.program2

Minimum Required SDK: API 14: Android 4.0 (IceCreamSandwich)

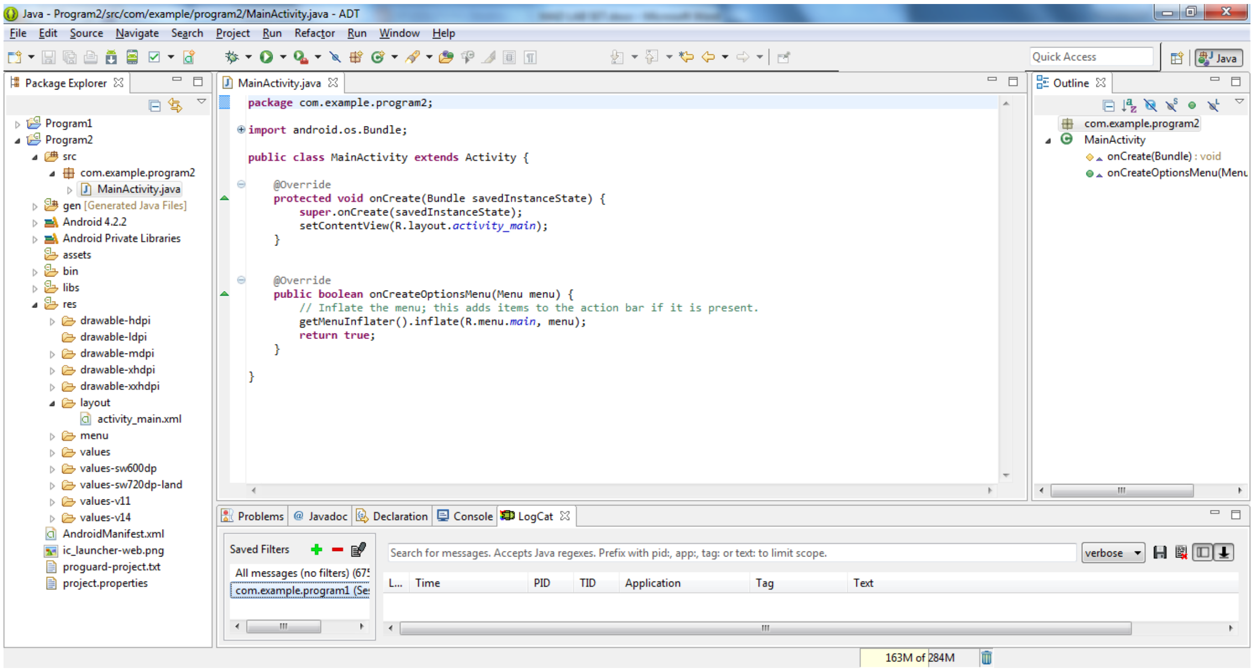
Target SDK: API 17: Android 4.2 (Jelly Bean)

Compile With: API 17: Android 4.2 (Jelly Bean)

Theme: Holo Light with Dark Action Bar

The application name is shown in the Play Store, as well as in the Manage Application list in Settings.

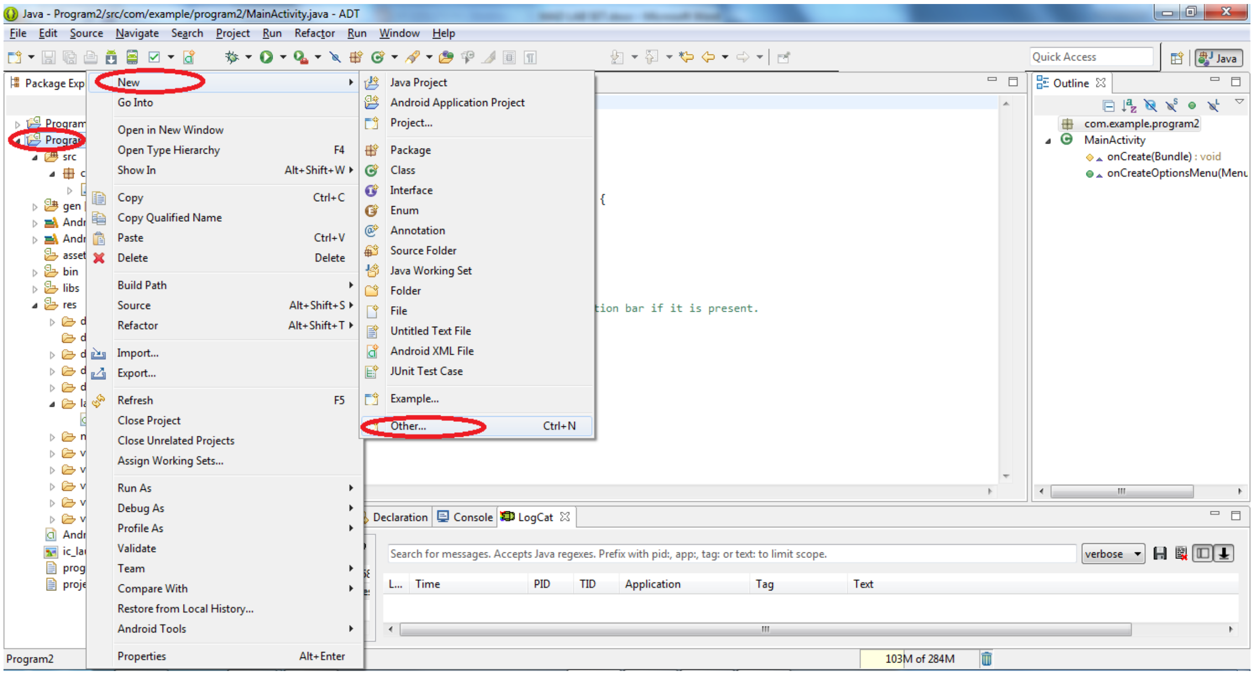
- Then click **Next.**
- Then click **Next.**
- Then select **Blank Activity** and click **Next.**
- Finally click F**inish**.
- It will take some time to build and to load the project.
- After completion it will look as given below.

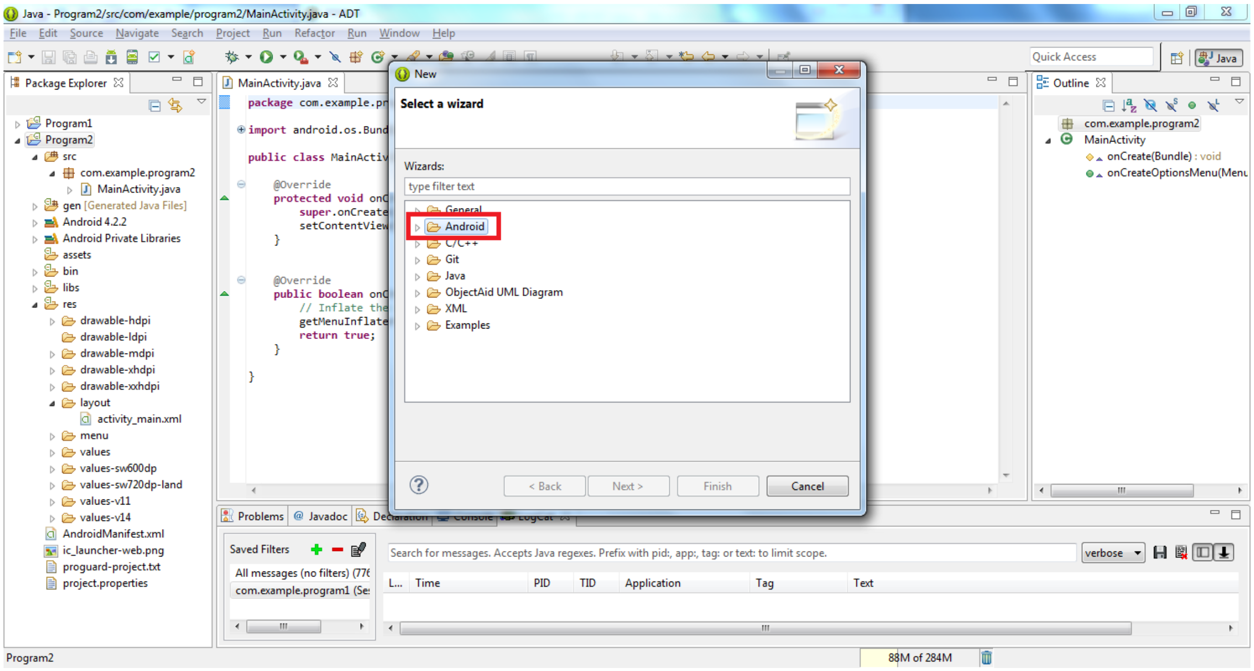


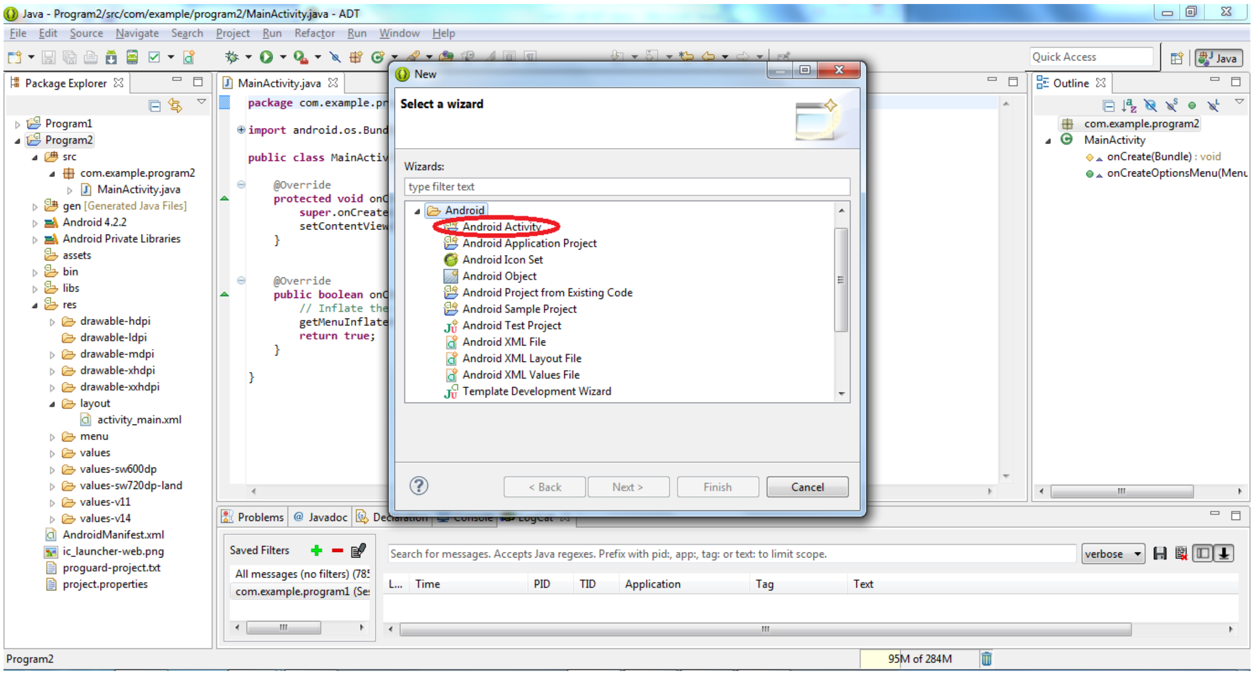
```
package com.example.program2;

import android.os.Bundle;

public class MainActivity extends Activity {

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

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

}
```

Creating Second Activity for the Android Application:

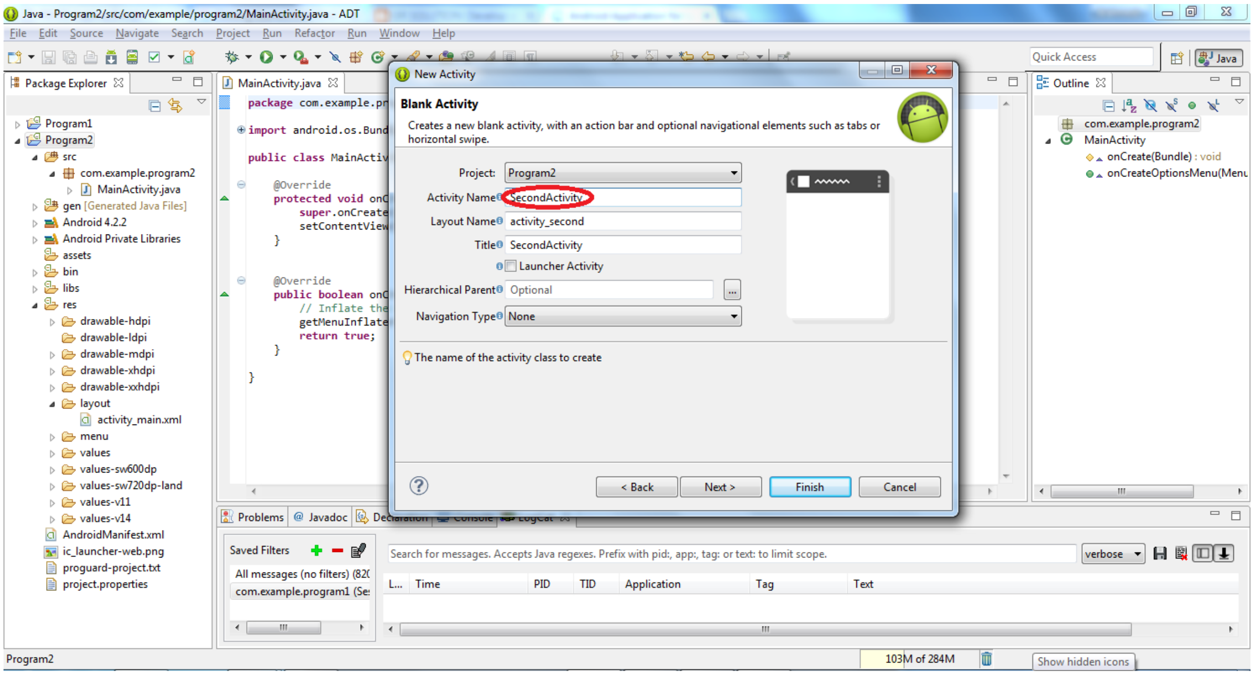
Click on Program2 -> New -> Other->Android->Android Activity.







- **Then Specify the Activity Name as SecondActivity and click Finish button.**



Blank Activity

Creates a new blank activity, with an action bar and optional navigational elements such as tabs or horizontal swipe.

Project: Program2

Activity Name: SecondActivity

Layout Name: activity_second

Title: SecondActivity

□ Launcher Activity

Hierarchical Parent: Optional

Navigation Type: None

The name of the activity class to create

```
package com.example.pr
import android.os.Bund
public class MainActiv
    @Override
    protected void onC
        super.onCreate
        setContentView
    }

    @Override
    public boolean onC
        // Inflate the
        getMenuInflate
        return true;
    }
}
```

Search for messages. Accepts Java regexes. Prefix with pid:, app:, tag: or text: to limit scope.

Designing layout for the Android Application:

Designing Layout for Main Activity:

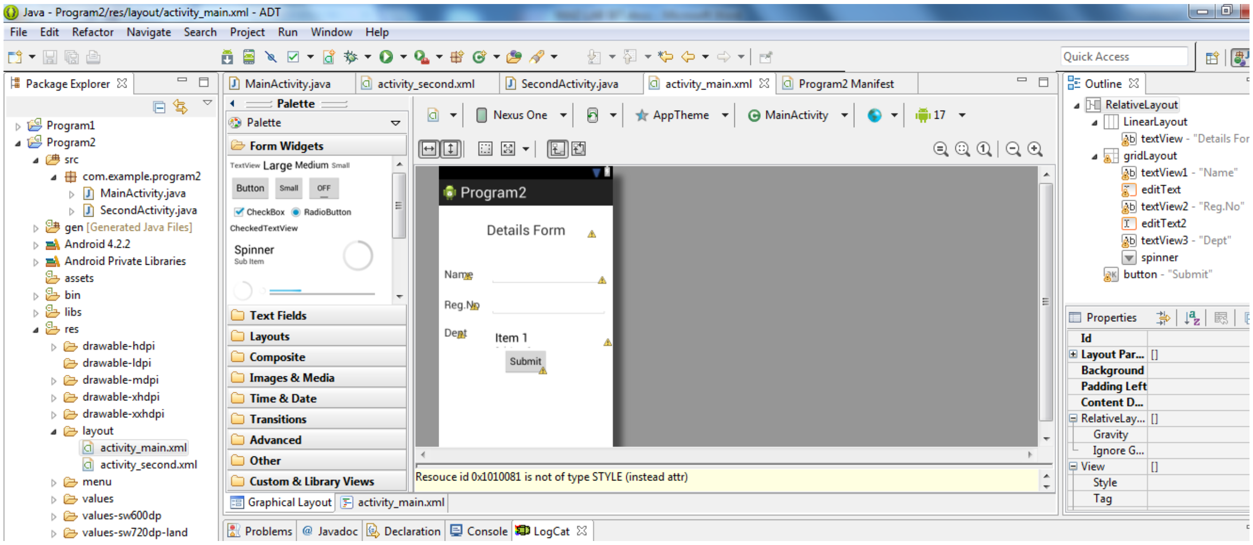
- Click on **Program2 -> res -> layout -> activity_main.xml.**
- Now go to XML code editor.
- Then delete the code which is there and type the code as given below.

Code for Activity_main.xml:

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

    <LinearLayout
```

```
        android:layout_width="match_parent"
        android:layout_height="100dp">
        <TextView
            android:id="@+id/textView"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="30dp"
            android:text="Details Form"
            android:textSize="25sp"
            android:gravity="center"/>
    </LinearLayout>

    <GridLayout
        android:id="@+id/gridLayout"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="100dp"
        android:layout_marginBottom="200dp"
        android:columnCount="2"
        android:rowCount="3">
        <TextView
            android:id="@+id/textView1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="10dp"
            android:layout_row="0"
            android:layout_column="0"
            android:text="Name"
            android:textSize="20sp"
            android:gravity="center"/>

        <EditText
            android:id="@+id/editText"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="10dp"
            android:layout_row="0"
            android:layout_column="1"
            android:ems="10"/>

        <TextView
            android:id="@+id/textView2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="10dp"
            android:layout_row="1"
            android:layout_column="0"
            android:text="Reg.No"
            android:textSize="20sp"
            android:gravity="center"/>

        <EditText
            android:id="@+id/editText2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="10dp"
            android:layout_row="1"
            android:layout_column="1"
            android:inputType="number"
            android:ems="10"/>

        <TextView
            android:id="@+id/textView3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="10dp"
            android:layout_row="2"
            android:layout_column="0"
            android:text="Dept"
            android:textSize="20sp"
            android:gravity="center"/>

        <Spinner
            android:id="@+id/spinner"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="10dp"
            android:layout_row="2"
            android:layout_column="1"
            android:spinnerMode="dropdown"/>

    </GridLayout>

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerInParent="true"
        android:layout_marginBottom="150dp"
        android:text="Submit"/>

</RelativeLayout>
```

- Now click on Design and your activity will look as given below.



Resouce id 0x1010081 is not of type STYLE (instead attr)

- So now the designing part of Main Activity is completed.

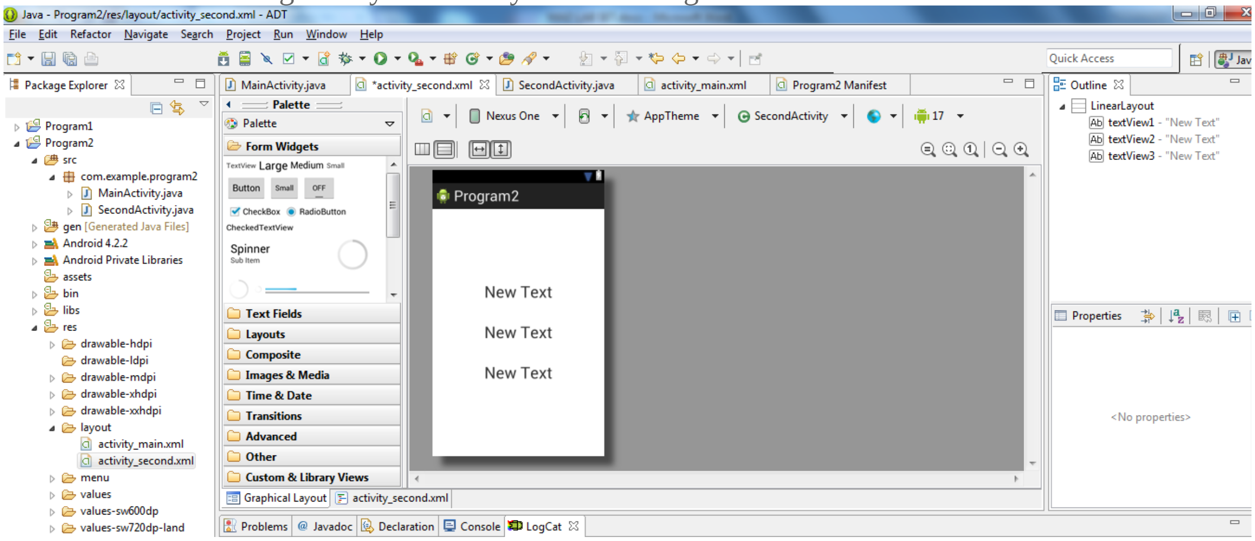
Designing Layout for Second Activity:

- Click on **app -> res -> layout -> activity_second.xml.**
- Now go to XML code editor.
- Then delete the code which is there and type the code as given below.

Code for Activity_second.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    tools:context="com.example.devang.exno2.SecondActivity"
    android:orientation="vertical"
    android:gravity="center">

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="20dp"
        android:text="New Text"
        android:textSize="30sp"/>

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="20dp"
        android:text="New Text"
        android:textSize="30sp"/>

    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="20dp"
        android:text="New Text"
        android:textSize="30sp"/>

</LinearLayout>
```

- Now click on Design and your activity will look as given below.



Java Coding for the Android Application:

Java Coidng for Main Activity:

- Click on **Program2 -> src -> com.example.program2 -> MainActivity.**
- Then delete the code which is there and type the code as given below.

Code for MainActivity.java:

```
package com.example.program2;

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

```
public class MainActivity extends Activity {

      //Defining the Views
    EditText e1,e2;
    Button bt;
    Spinner s;

    //Data for populating in Spinner
    String [] dept_array={"ISE","ECE","CSE","Mech","Civil"};

    String name,reg,dept;

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

        //Referring the Views
        e1= (EditText) findViewById(R.id.editText);
        e2= (EditText) findViewById(R.id.editText2);

        bt= (Button) findViewById(R.id.button);

        s= (Spinner) findViewById(R.id.spinner);

        //Creating Adapter for Spinner for adapting the data from array to Spinner
        ArrayAdapter adapter= new
ArrayAdapter(MainActivity.this,android.R.layout.simple_spinner_item,dept_array);
        s.setAdapter(adapter);

        //Creating Listener for Button
        bt.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                //Getting the Values from Views(Edittext & Spinner)
                name=e1.getText().toString();
                reg=e2.getText().toString();
                dept=s.getSelectedItem().toString();

                //Intent For Navigating to Second Activity
                Intent i = new Intent(MainActivity.this,SecondActivity.class);

                //For Passing the Values to Second Activity
                i.putExtra("name_key", name);
                i.putExtra("reg_key",reg);
                i.putExtra("dept_key", dept);

                startActivity(i);

            }
        });
    }

}
```

- So now the Coding part of Main Activity is completed.

Java Coding for Second Activity:

- Click on **Program2 -> src -> com.example.program2 -> SecondActivity.**
- Then delete the code which is there and type the code as given below.

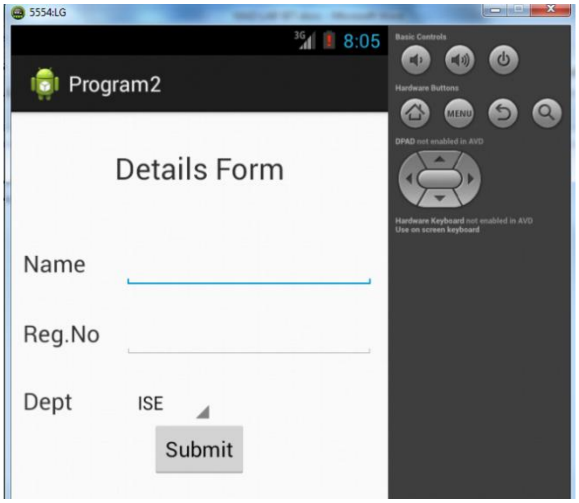
Code for SecondActivity.java:

```
package com.example.program2;

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

import android.widget.TextView;

public class SecondActivity extends Activity {

       TextView t1,t2,t3;

    String name,reg,dept;

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

        t1= (TextView) findViewById(R.id.textView1);
        t2= (TextView) findViewById(R.id.textView2);
        t3= (TextView) findViewById(R.id.textView3);

        //Getting the Intent
        Intent i = getIntent();

        //Getting the Values from First Activity using the Intent received
        name=i.getStringExtra("name_key");
        reg=i.getStringExtra("reg_key");
        dept=i.getStringExtra("dept_key");

        //Setting the Values to Intent
        t1.setText(name);
        t2.setText(reg);
        t3.setText(dept);

    }

}
```

- So now the Coding part of Second Activity is also completed.

- Now run the application to see the output.

Output:



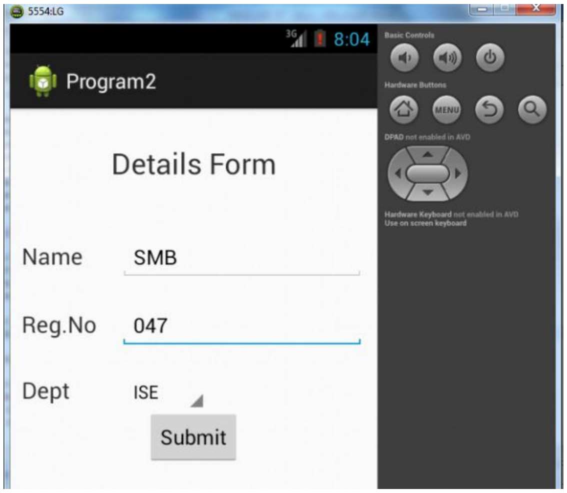
Program2

Details Form

Name

Reg.No

Dept ISE



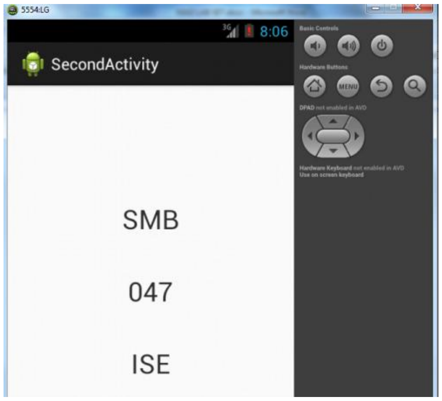
Program2

Details Form

Name SMB

Reg.No 047

Dept ISE



SecondActivity

SMB

047

ISE

Result:

Thus a Simple Android Application that uses Layout Managers and Event Listeners is developed and executed successfully.

PROGRAM 3

Aim:

To develop a Simple Android Application for Native Calculator.

Procedure:

Designing layout for the Android Application:

Designing Layout for Main Activity:

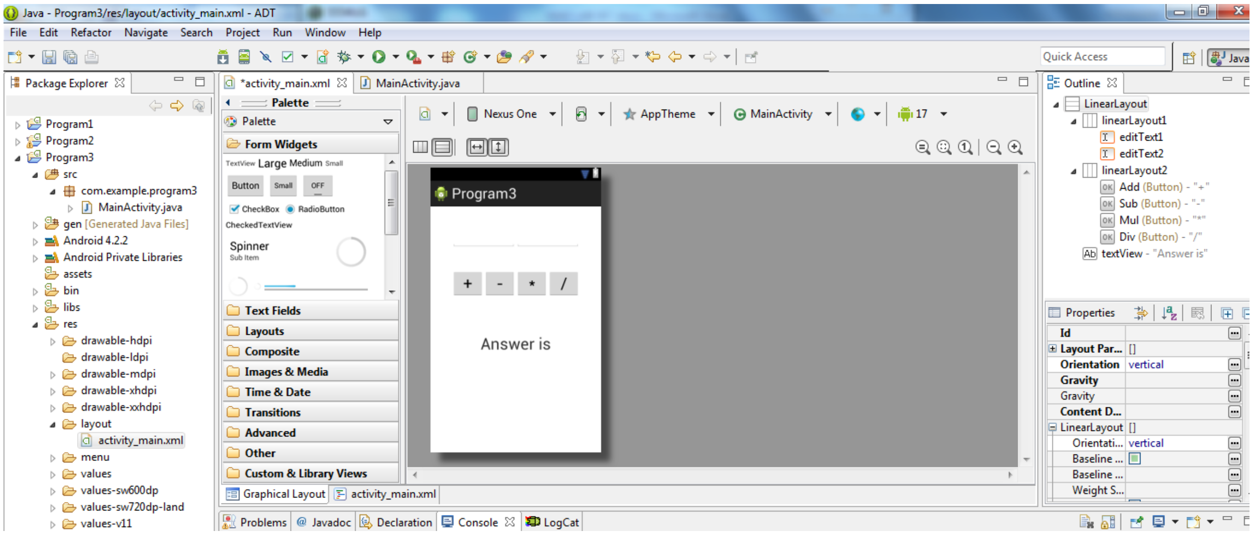
- Click on **Program3 -> res -> layout -> activity_main.xml.**
- Now go to XML code editor.
- Then delete the code which is there and type the code as given below.

Code for Activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="20dp">

    <LinearLayout
        android:id="@+id/linearLayout1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="20dp">

        <EditText
            android:id="@+id/editText1"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
        android:inputType="numberDecimal"
        android:textSize="20sp" />

    <EditText
        android:id="@+id/editText2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:inputType="numberDecimal"
        android:textSize="20sp" />

</LinearLayout>

<LinearLayout
    android:id="@+id/linearLayout2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="20dp">

    <Button
        android:id="@+id/Add"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="+"
        android:textSize="30sp"/>

    <Button
        android:id="@+id/Sub"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="-"
        android:textSize="30sp"/>

    <Button
        android:id="@+id/Mul"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="*"
        android:textSize="30sp"/>

    <Button
        android:id="@+id/Div"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="/"
        android:textSize="30sp"/>

</LinearLayout>

<TextView
    android:id="@+id/textView"
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="50dp"
        android:text="Answer is"
        android:textSize="30sp"
        android:gravity="center"/>

</LinearLayout>
```

- Now click on Design and your activity will look as given below.



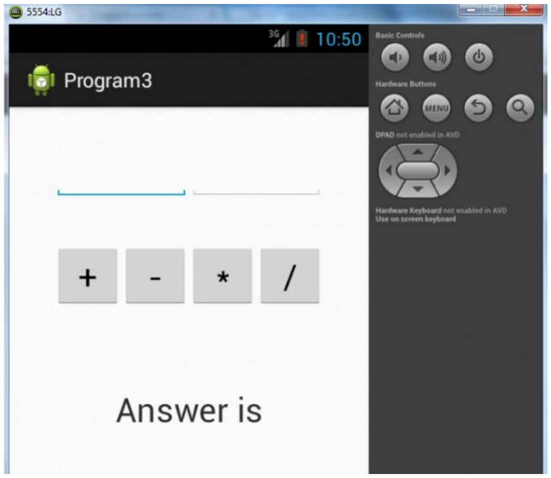
- So now the designing part is completed.

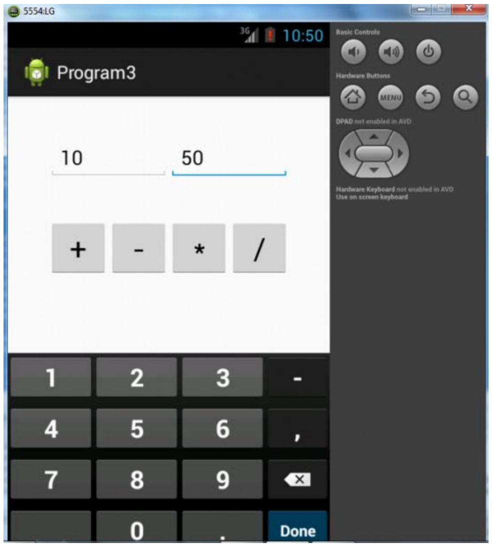
Java Coding for the Android Application:

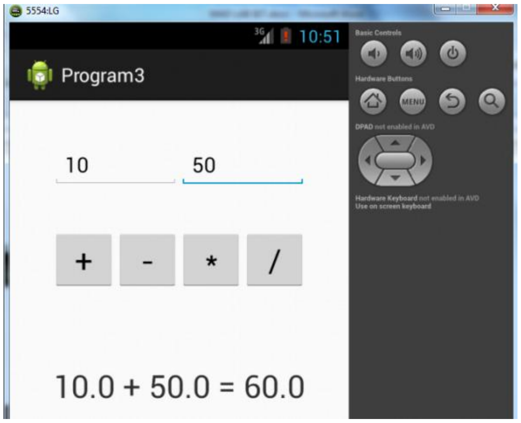
- Click on **Program3 ->src -> com.example.Program3 -> MainActivity.**

```
package com.example.program3;

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

public class MainActivity extends Activity implements OnClickListener{


    //Defining the Views
     EditText Num1;
     EditText Num2;
     Button Add;
     Button Sub;
     Button Mul;
     Button Div;
    TextView Result;

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

        //Referring the Views
        Num1 = (EditText) findViewById(R.id.editText1);
        Num2 = (EditText) findViewById(R.id.editText2);
        Add = (Button) findViewById(R.id.Add);
        Sub = (Button) findViewById(R.id.Sub);
        Mul = (Button) findViewById(R.id.Mul);
        Div = (Button) findViewById(R.id.Div);
        Result = (TextView) findViewById(R.id.textView);

        // set a listener
        Add.setOnClickListener(this);
        Sub.setOnClickListener(this);
        Mul.setOnClickListener(this);
        Div.setOnClickListener(this);
    }

    @Override
    public void onClick (View v)
    {

        float num1 = 0;
        float num2 = 0;
        float result = 0;
        String oper = "";

        // check if the fields are empty
        if (TextUtils.isEmpty(Num1.getText().toString()) ||
TextUtils.isEmpty(Num2.getText().toString()))
                return;

        // read EditText and fill variables with numbers
        num1 = Float.parseFloat(Num1.getText().toString());
        num2 = Float.parseFloat(Num2.getText().toString());

        // defines the button that has been clicked and performs the
corresponding operation
        // write operation into oper, we will use it later for output
        switch (v.getId())
        {
            case R.id.Add:
                oper = "+";
                result = num1 + num2;
                break;
            case R.id.Sub:
                oper = "-";
                result = num1 - num2;
                break;
            case R.id.Mul:
                oper = "*";
                result = num1 * num2;
                break;
            case R.id.Div:
                oper = "/";
                result = num1 / num2;
                break;
            default:
                break;
        }
        // form the output line
        Result.setText(num1 + " " + oper + " " + num2 + " = " + result);
    }

}
```

- So now the Coding part is also completed.
- Now run the application to see the output.







```
10.0 + 50.0 = 60.0
```

Result:

Thus a Simple Android Application for Native Calculator is developed and executed successfully.

PROGRAM 4

Aim:

To develop a Simple Android Application that draws basic Graphical Primitives on the screen.

Procedure:

Designing layout for the Android Application:

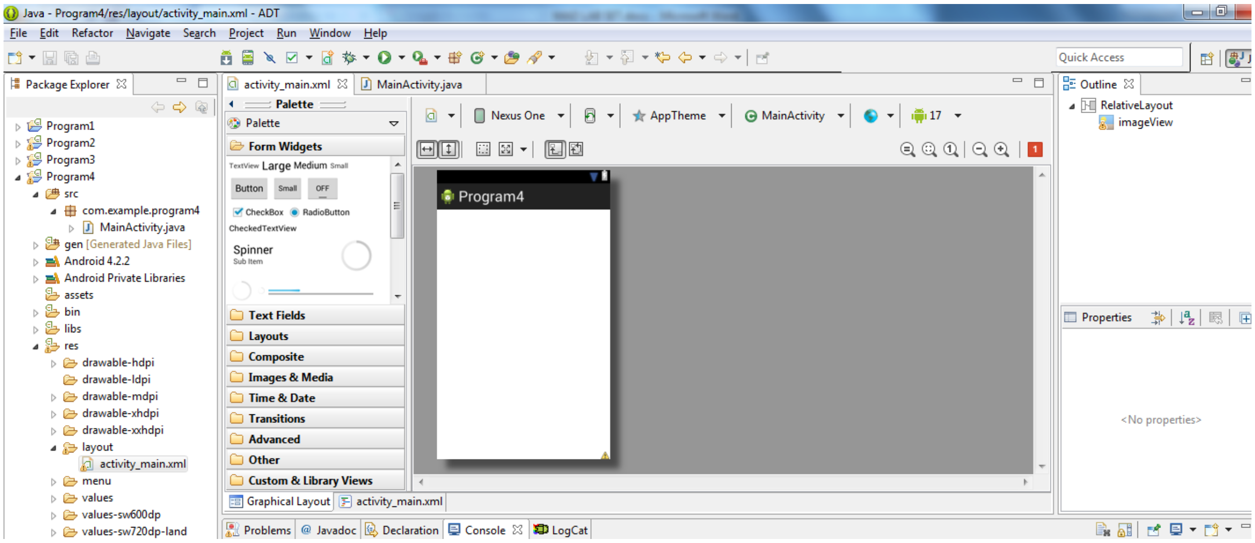
Designing Layout for Main Activity:

- Click on **Program4 -> res -> layout -> activity_main.xml.**
- Now go to XML code editor.
- Then delete the code which is there and type the code as given below.

Code for Activity_main.xml:

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

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/imageView" />
</RelativeLayout>
```

- Now click on Design and your activity will look as given below.



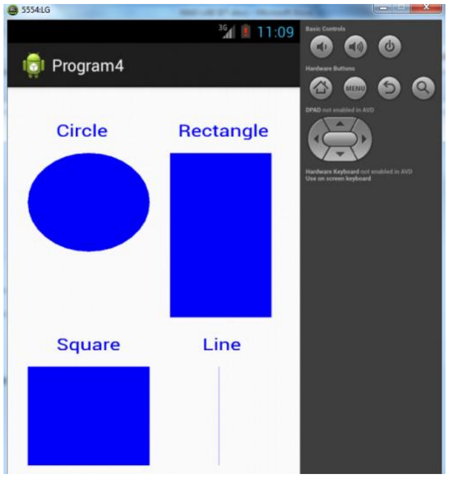
- So now the designing part is completed.

Java Coding for the Android Application:

Click on **Program4 ->src -> com.example.Program4 -> MainActivity.**

```
package com.example.program4;

import android.os.Bundle;
import android.app.Activity;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.drawable.BitmapDrawable;
import android.widget.ImageView;

public class MainActivity extends Activity {



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

        //Creating a Bitmap
        Bitmap bg = Bitmap.createBitmap(720, 1280, Bitmap.Config.ARGB_8888);

        //Setting the Bitmap as background for the ImageView
        ImageView i = (ImageView) findViewById(R.id.imageView);
        i.setBackgroundDrawable(new BitmapDrawable(bg));

        //Creating the Canvas Object
        Canvas canvas = new Canvas(bg);

        //Creating the Paint Object and set its color & TextSize
        Paint paint = new Paint();
        paint.setColor(Color.BLUE);
        paint.setTextSize(50);

        //To draw a Rectangle
        canvas.drawText("Rectangle", 420, 150, paint);
        canvas.drawRect(400, 200, 650, 700, paint);

        //To draw a Circle
        canvas.drawText("Circle", 120, 150, paint);
        canvas.drawCircle(200, 350, 150, paint);

        //To draw a Square
        canvas.drawText("Square", 120, 800, paint);
        canvas.drawRect(50, 850, 350, 1150, paint);

        //To draw a Line
        canvas.drawText("Line", 480, 800, paint);
        canvas.drawLine(520, 850, 520, 1150, paint);
    }

}
```

- So now the Coding part is also completed.
- Now run the application to see the output.



Result:

Thus a Simple Android Application that draws basic Graphical Primitives on the screen is developed and executed successfully.

PROGRAM 5

Aim:

To develop a Simple Android Application that makes use of Database.

Procedure:

Designing layout for the Android Application:

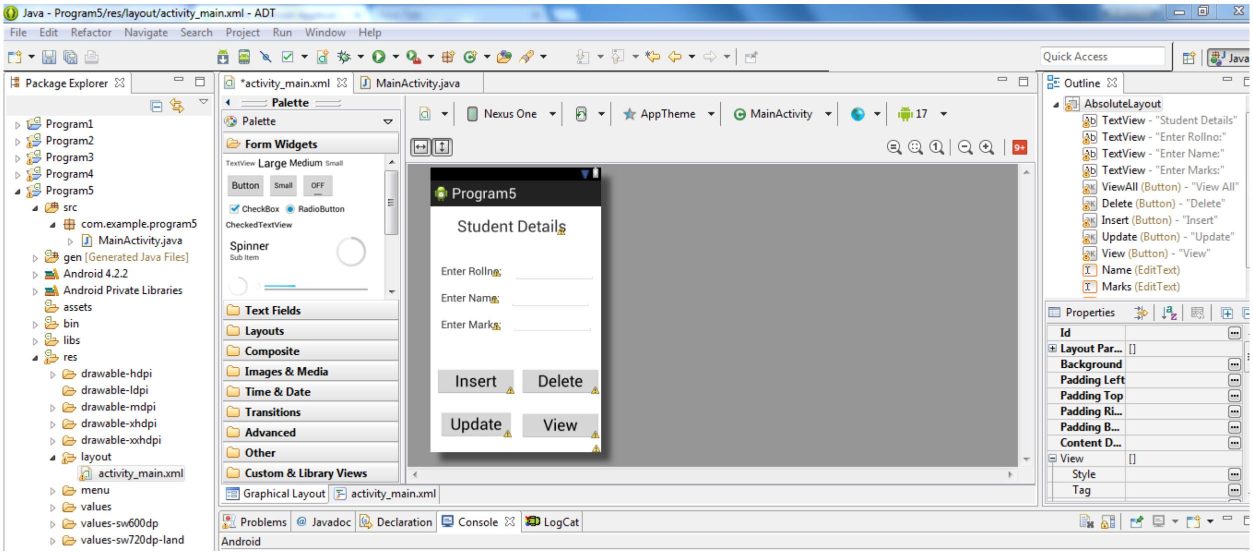
Designing Layout for Main Activity:

- Click on **Program5 -> res -> layout -> activity_main.xml.**
- Now go to XML code editor.
- Then delete the code which is there and type the code as given below.

Code for Activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_x="50dp"
        android:layout_y="20dp"
        android:text="Student Details"
        android:textSize="30sp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_x="20dp"
        android:layout_y="82dp"
        android:text="Enter Rollno:"
        android:textSize="20sp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_x="24dp"
        android:layout_y="131dp"
        android:text="Enter Name:"
        android:textSize="20sp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_x="26dp"
        android:layout_y="181dp"
        android:text="Enter Marks:"
        android:textSize="20sp" />

    <EditText
        android:id="@+id/Name"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_x="152dp"
        android:layout_y="126dp"
        android:ems="10"
        android:inputType="text"
        android:textSize="20sp" />

    <EditText
        android:id="@+id/Marks"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_x="155dp"
        android:layout_y="172dp"
        android:ems="10"
        android:inputType="number"
        android:textSize="20sp" />

    <EditText
        android:id="@+id/Rollno"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_x="152dp"
        android:layout_y="77dp"
        android:ems="10"
        android:inputType="number"
        android:textSize="20sp" >

        <requestFocus />
    </EditText>

    <Button
        android:id="@+id/Insert"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_x="10dp"
        android:layout_y="256dp"
        android:text="Insert"
        android:textSize="30dp" />

    <Button
        android:id="@+id/Delete"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_x="173dp"
        android:layout_y="257dp"
        android:text="Delete"
        android:textSize="30dp" />

    <Button
        android:id="@+id/Update"
        android:layout_width="138dp"
        android:layout_height="wrap_content"
        android:layout_x="15dp"
        android:layout_y="335dp"
        android:text="Update"
        android:textSize="30dp" />

    <Button
        android:id="@+id/View"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_x="171dp"
        android:layout_y="338dp"
        android:text="View"
        android:textSize="30dp" />

    <Button
        android:id="@+id/ViewAll"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:layout_x="65dp"
        android:layout_y="401dp"
        android:text="View All"
        android:textSize="30dp" />

</AbsoluteLayout>
```

- Now click on Design and your activity will look as given below.



- So now the designing part is completed.

Java Coding for the Android Application:

Click on **Program5 ->src -> com.example.Program5 -> MainActivity.**

```
package com.example.program5;

import android.os.Bundle;
import android.app.Activity;
import android.content.Context;
import android.app.AlertDialog.Builder;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends Activity implements OnClickListener {

    EditText Rollno,Name,Marks;
    Button Insert,Delete,Update,View,ViewAll;
    SQLiteDatabase db;
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Rollno=(EditText)findViewById(R.id.Rollno);
        Name=(EditText)findViewById(R.id.Name);
        Marks=(EditText)findViewById(R.id.Marks);
        Insert=(Button)findViewById(R.id.Insert);
        Delete=(Button)findViewById(R.id.Delete);
        Update=(Button)findViewById(R.id.Update);
        View=(Button)findViewById(R.id.View);
        ViewAll=(Button)findViewById(R.id.ViewAll);

        Insert.setOnClickListener(this);
        Delete.setOnClickListener(this);
        Update.setOnClickListener(this);
        View.setOnClickListener(this);
        ViewAll.setOnClickListener(this);

        // Creating database and table
        db=openOrCreateDatabase("StudentDB", Context.MODE_PRIVATE, null);
        db.execSQL("CREATE TABLE IF NOT EXISTS student(rollno VARCHAR,name
VARCHAR,marks VARCHAR);");
    }
    public void onClick(android.view.View view)
    {
        // Inserting a record to the Student table
        if(view==Insert)
        {
            // Checking for empty fields
            if(Rollno.getText().toString().trim().length()==0||
                    Name.getText().toString().trim().length()==0||
                    Marks.getText().toString().trim().length()==0)
            {
                showMessage("Error", "Please enter all values");
                return;
            }
            db.execSQL("INSERT INTO student
VALUES('"+Rollno.getText()+"','"+Name.getText()+
                    "','"+Marks.getText()+"');");
            showMessage("Success", "Record added");
            clearText();
        }
        // Deleting a record from the Student table
        if(view==Delete)
        {
            // Checking for empty roll number
            if(Rollno.getText().toString().trim().length()==0)
            {
                showMessage("Error", "Please enter Rollno");
                return;
            }
            Cursor c=db.rawQuery("SELECT * FROM student WHERE
rollno='"+Rollno.getText()+"'", null);
            if(c.moveToFirst())
            {
                db.execSQL("DELETE FROM student WHERE
rollno='"+Rollno.getText()+"'");
                showMessage("Success", "Record Deleted");
            }
            else
            {
                showMessage("Error", "Invalid Rollno");
            }
            clearText();
        }
        // Updating a record in the Student table
        if(view==Update)
        {
            // Checking for empty roll number
            if(Rollno.getText().toString().trim().length()==0)
            {
                showMessage("Error", "Please enter Rollno");
                return;
            }
            Cursor c=db.rawQuery("SELECT * FROM student WHERE
rollno='"+Rollno.getText()+"'", null);
            if(c.moveToFirst()) {
                db.execSQL("UPDATE student SET name='" + Name.getText() + "',marks='"
+ Marks.getText() +
                        "' WHERE rollno='"+Rollno.getText()+"'");
                showMessage("Success", "Record Modified");
            }
            else {
                showMessage("Error", "Invalid Rollno");
            }
            clearText();
        }
        // Display a record from the Student table
        if(view==View)
        {
            // Checking for empty roll number
            if(Rollno.getText().toString().trim().length()==0)
            {
                showMessage("Error", "Please enter Rollno");
                return;
            }
            Cursor c=db.rawQuery("SELECT * FROM student WHERE
rollno='"+Rollno.getText()+"'", null);
            if(c.moveToFirst())
            {
                Name.setText(c.getString(1));
                Marks.setText(c.getString(2));
            }
            else
            {
                showMessage("Error", "Invalid Rollno");
                clearText();
            }
        }
        // Displaying all the records
        if(view==ViewAll)
        {
            Cursor c=db.rawQuery("SELECT * FROM student", null);
            if(c.getCount()==0)
            {
                showMessage("Error", "No records found");
                return;
            }
            StringBuffer buffer=new StringBuffer();
            while(c.moveToNext())
            {
                buffer.append("Rollno: "+c.getString(0)+"\n");
                buffer.append("Name: "+c.getString(1)+"\n");
                buffer.append("Marks: "+c.getString(2)+"\n\n");
            }
            showMessage("Student Details", buffer.toString());
        }
    }
    public void showMessage(String title,String message)
    {
        Builder builder=new Builder(this);
        builder.setCancelable(true);
        builder.setTitle(title);
        builder.setMessage(message);
        builder.show();
    }
    public void clearText()
    {
        Rollno.setText("");
        Name.setText("");
        Marks.setText("");
        Rollno.requestFocus();
    }


}
```

- So now the Coding part is also completed.
- Now run the application to see the output.

Output:

Getting Exception Check

PROGRAM 10

Aim:

To develop an application that creates an alert upon receiving a message.

Procedure:

Create Main Activity for the Android Application

Minimim Required SDK = API 16 and click **Next.**

Create Second Activity for the Android Application

Designing layout for the Android Application:

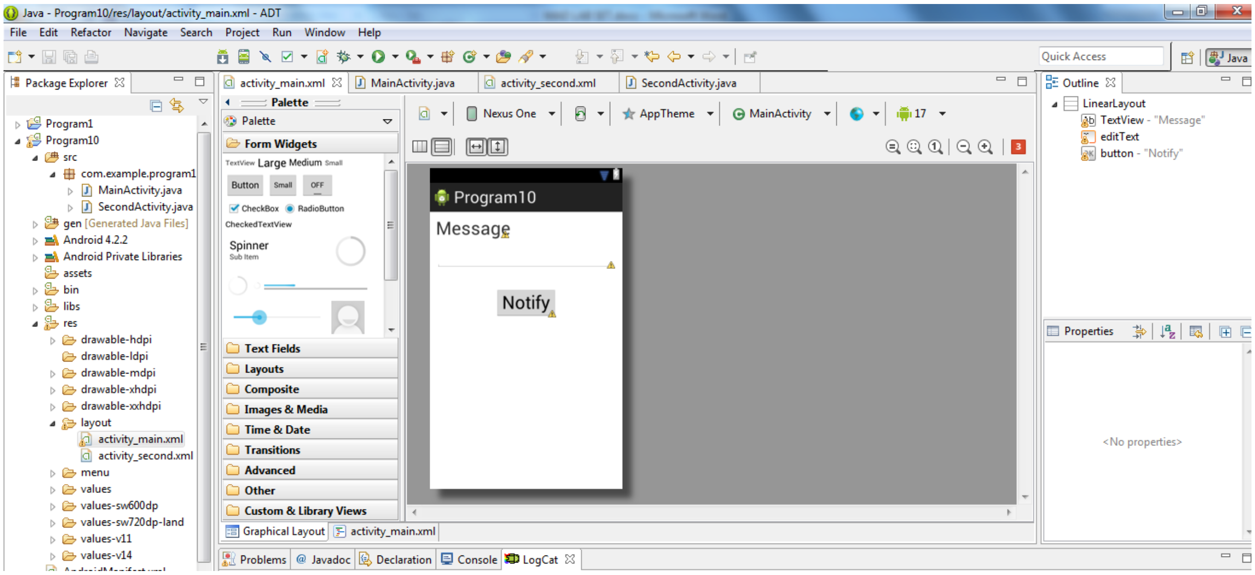
Designing Layout for Main Activity:

- Click on **Program10 -> res -> layout -> activity_main.xml.**
- Now go to XML code editor.
- Then delete the code which is there and type the code as given below.

Code for 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:layout_margin="10dp"
    android:orientation="vertical">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Message"
        android:textSize="30sp" />

    <EditText
        android:id="@+id/editText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:singleLine="true"
        android:textSize="30sp" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="30dp"
        android:layout_gravity="center"
        android:text="Notify"
        android:textSize="30sp"/>

</LinearLayout>
```

- Now click on Design and your activity will look as given below.



- So now the designing part is completed.

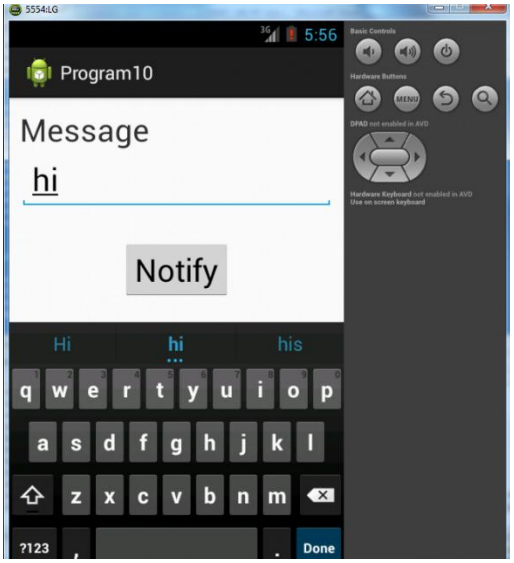
Java Coding for the Android Application:

Click on **Program10 ->src -> com.example.Program10 -> MainActivity.**

```
package com.example.program10;

import android.os.Bundle;
import android.app.Activity;
import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;


public class MainActivity extends Activity implements OnClickListener{

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

        notify= (Button) findViewById(R.id.button);
        e= (EditText) findViewById(R.id.editText);

        notify.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v)
            {
                Intent intent = new Intent(MainActivity.this, SecondActivity.class);
                PendingIntent pending = PendingIntent.getActivity(MainActivity.this,
0, intent, 0);
                Notification noti = new
Notification.Builder(MainActivity.this).setContentTitle("New
Message").setContentText(e.getText().toString()).setSmallIcon(R.drawable.ic_launcher)
.setContentIntent(pending).build();
                NotificationManager manager = (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);
                noti.flags |= Notification.FLAG_AUTO_CANCEL;
                manager.notify(0, noti);
            }
        });
    }
      @Override
      public void onClick(View arg0) {
             // TODO Auto-generated method stub

      }

}
```

- So now the Coding part is also completed.
- Now run the application to see the output.

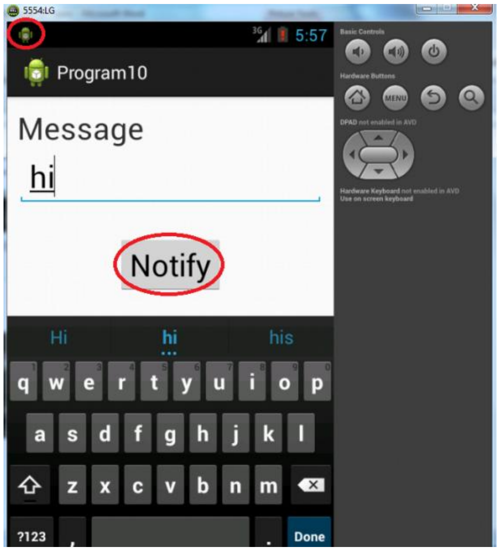
Output:



Program10

Message

hi



Result:

Thus Android Application that creates an alert upon receiving a message is developed and executed successfully.

PROGRAM 11

Aim:

To develop a mobile application that creates alarm clock.

Procedure:

Create Main Activity for the Android Application

Minimim Required SDK = API 16 and click **Next.**

Create Second Activity for the Android Application

Designing layout for the Android Application:

Designing Layout for Main Activity:

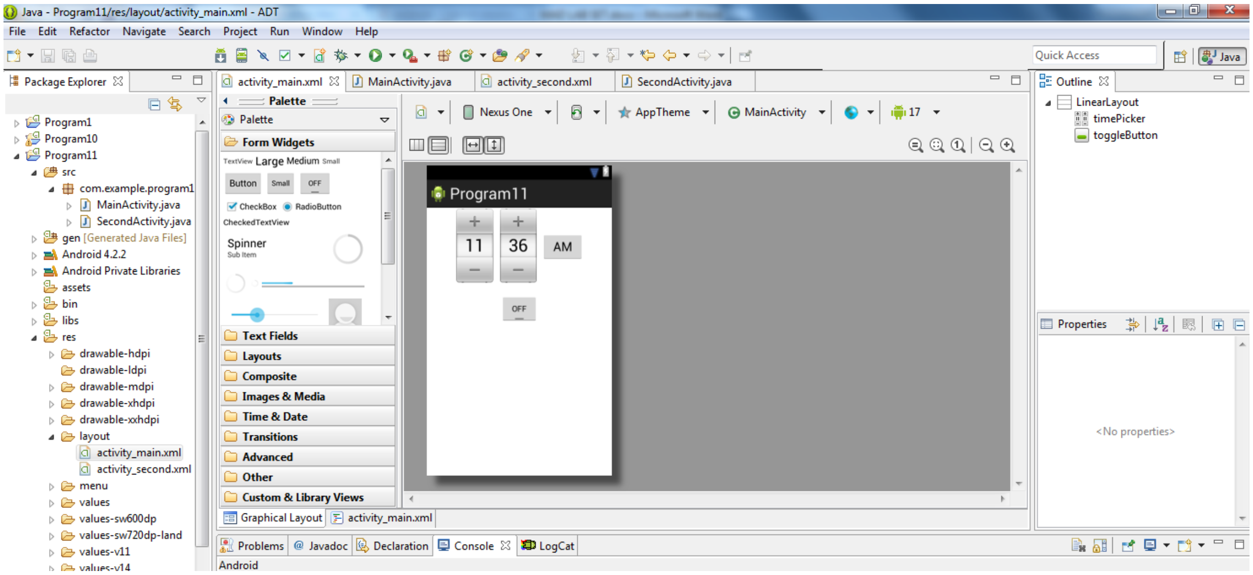
- Click on **Program11 -> res -> layout -> activity_main.xml.**
- Now go to XML code editor.

- Then delete the code which is there and type the code as given below.

Code for 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">

    <TimePicker
        android:id="@+id/timePicker"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center" />

    <ToggleButton
        android:id="@+id/toggleButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_margin="20dp"
        android:checked="false"
        android:onClick="OnToggleClicked" />

</LinearLayout>
```

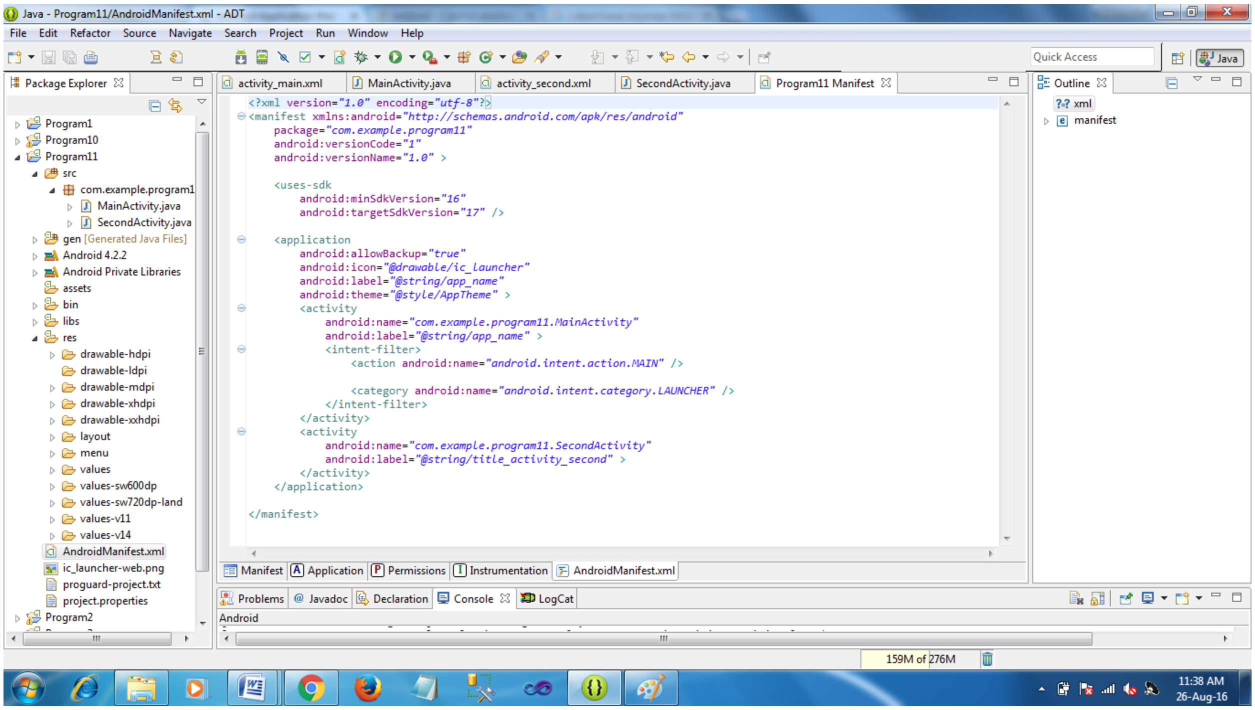
- Now click on **Design** and your application will look as given below.



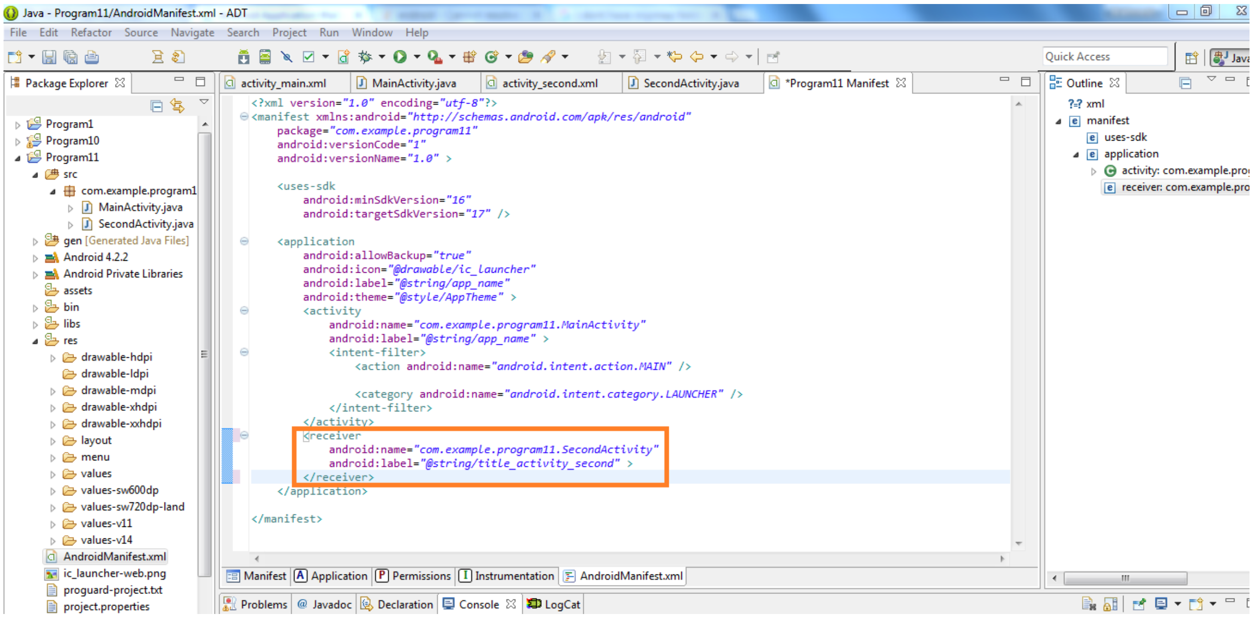
- So now the designing part is completed.

Changes in Manifest for the Android Application:

- Click on **Program11 -> manifests -> AndroidManifest.xml**



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.program11"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="16"
        android:targetSdkVersion="17" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.example.program11.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="com.example.program11.SecondActivity"
            android:label="@string/title_activity_second" >
        </activity>
    </application>

</manifest>
```



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.program11"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="16"
        android:targetSdkVersion="17" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.example.program11.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>
        <receiver
            android:name="com.example.program11.SecondActivity"
            android:label="@string/title_activity_second" >
        </receiver>
    </application>

</manifest>
```

Java Coding for the Android Application:

Java Coding for Main Activity:

- Click on **Program11 -> java -> com.example.Program11 -> MainActivity.**

Code for MainActivity.java

```java
package com.example.program11;

import java.util.Calendar;
import android.widget.ToggleButton;

import android.os.Bundle;
import android.app.Activity;
import android.app.AlarmManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.view.Menu;
import android.view.View;
import android.widget.TimePicker;
import android.widget.Toast;

public class MainActivity extends Activity {

      TimePicker alarmTimePicker;
    PendingIntent pendingIntent;
    AlarmManager alarmManager;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        alarmTimePicker = (TimePicker) findViewById(R.id.timePicker);
        alarmManager = (AlarmManager) getSystemService(ALARM_SERVICE);
    }
    public void OnToggleClicked(View view)
    {
        long time;
        if (((ToggleButton) view).isChecked())
        {
            Toast.makeText(MainActivity.this, "ALARM ON", Toast.LENGTH_SHORT).show();
            Calendar calendar = Calendar.getInstance();
            calendar.set(Calendar.HOUR_OF_DAY, alarmTimePicker.getCurrentHour());
            calendar.set(Calendar.MINUTE, alarmTimePicker.getCurrentMinute());
            Intent intent = new Intent(this, SecondActivity.class);
            pendingIntent = PendingIntent.getBroadcast(this, 0, intent, 0);

            time=(calendar.getTimeInMillis()-(calendar.getTimeInMillis()%60000));
            if(System.currentTimeMillis()>time)
            {
                if (calendar.AM_PM == 0)
                    time = time + (1000*60*60*12);
                else
```

```
                    time = time + (1000*60*60*24);
            }
            alarmManager.setRepeating(AlarmManager.RTC_WAKEUP, time, 10000,
pendingIntent);
        }
        else
        {
            alarmManager.cancel(pendingIntent);
            Toast.makeText(MainActivity.this, "ALARM OFF",
Toast.LENGTH_SHORT).show();
        }
    }

}
```

- So now the Coding part of Main Activity is completed.

Java Coding for Second Activity:

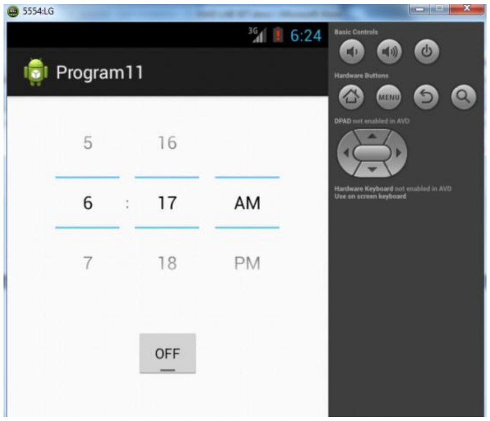
- Click on **Program11 -> java -> com.example.Program11 -> SecondActivity.**

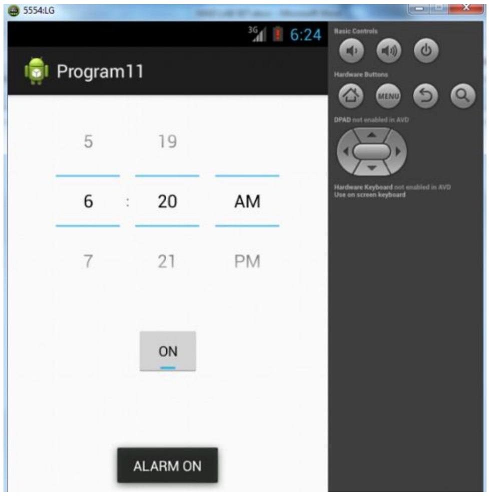
Code for SecondActivity.java:

```
package com.example.program11;

import android.media.Ringtone;
import android.content.BroadcastReceiver;
import android.media.RingtoneManager;
import android.net.Uri;

import android.content.Context;
import android.content.Intent;

import android.widget.Toast;

public class SecondActivity extends BroadcastReceiver {

       @Override
          public void onReceive(Context context, Intent intent)
          {
              Toast.makeText(context, "Alarm! Wake up! Wake up!",
Toast.LENGTH_LONG).show();
              Uri alarmUri =
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_ALARM);
              if (alarmUri == null)
              {
                  alarmUri =
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
              }
              Ringtone ringtone = RingtoneManager.getRingtone(context, alarmUri);
              ringtone.play();
          }
}
```

- So now the Coding part of Alarm Receiver is also completed.
- Now run the application to see the output.

Output:





Result:

Thus Android Application that creates Alarm Clock is developed and executed successfully.

PROGRAM 9

Aim:

To develop an Android Application that writes data to the SD Card.

Procedure:

Create Main Activity for the Android Application

Minimim Required SDK = API 15 and click **Next.**

Designing layout for the Android Application:

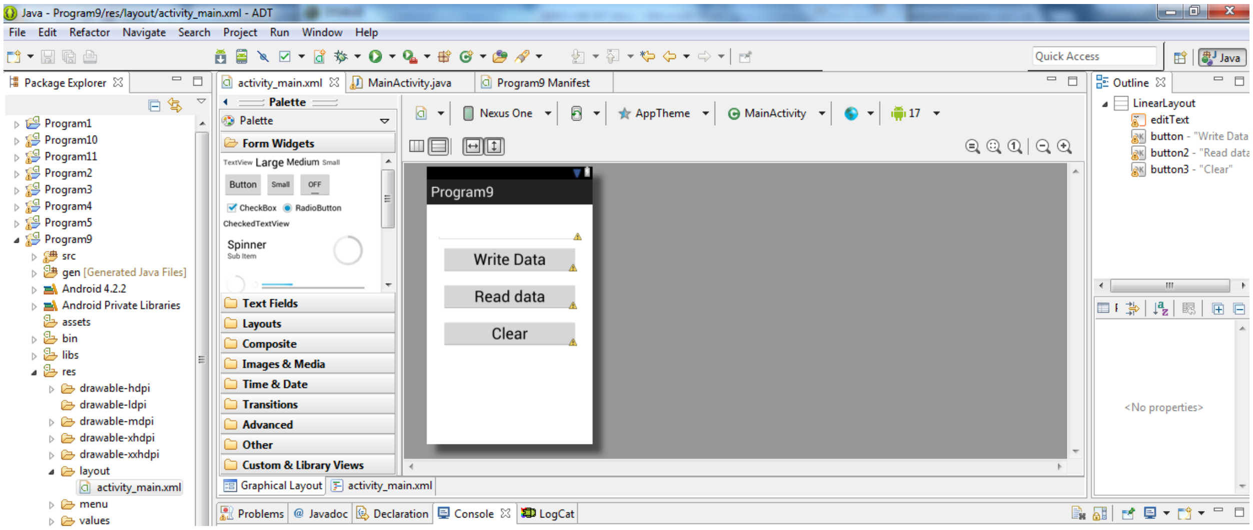
Designing Layout for Main Activity:

- Click on **Program9 -> res -> layout -> activity_main.xml.**
- Now go to XML code editor.
- Then delete the code which is there and type the code as given below.

Code for 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:layout_margin="20dp"
    android:orientation="vertical">

    <EditText
        android:id="@+id/editText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:singleLine="true"
        android:textSize="30dp" />

    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:text="Write Data"
        android:textSize="30dp" />

    <Button
        android:id="@+id/button2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:text="Read data"
        android:textSize="30dp" />

    <Button
        android:id="@+id/button3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:text="Clear"
        android:textSize="30dp" />

</LinearLayout>
```

- Now click on **Design** and your application will look as given below.



- So now the designing part is completed.

Changes in Manifest for the Android Application:

- Click on **Program9 -> manifests -> AndroidManifest.xml**
  - Now go to XML code editor.
  - Then delete the code which is there and type the code as given below.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.program9"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="15"
        android:targetSdkVersion="17" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>

    <application
        android:allowBackup="true"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme" >
        <activity android:name=".MainActivity" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

Java Coding for the Android Application:

Java Coding for Main Activity:

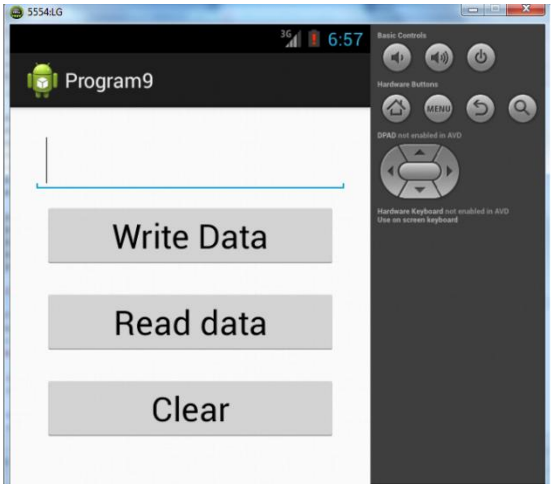
- Click on **Program9 -> java -> com.example.Program9 -> MainActivity.**

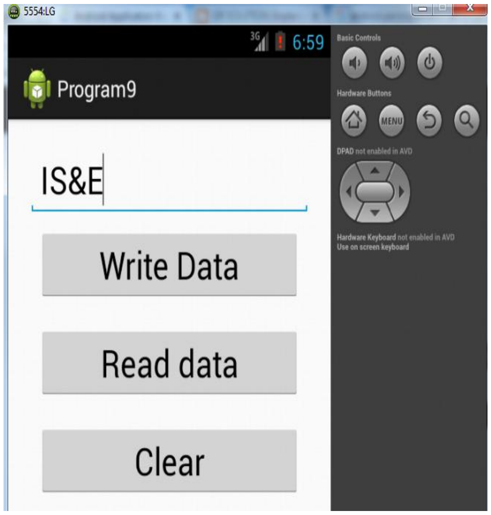
Code for MainActivity.java

```
package com.example.program9;

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

import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStreamReader;

public class MainActivity extends Activity {

      EditText e1;
    Button write,read,clear;
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        e1= (EditText) findViewById(R.id.editText);
        write= (Button) findViewById(R.id.button);
        read= (Button) findViewById(R.id.button2);
        clear= (Button) findViewById(R.id.button3);

        write.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v)
            {
                String message=e1.getText().toString();
                try
                {
                    File f=new File("/sdcard/myfile.txt");
                    f.createNewFile();
                    FileOutputStream fout=new FileOutputStream(f);
                    fout.write(message.getBytes());
                    fout.close();
                    Toast.makeText(getBaseContext(),"Data Written in
SDCARD",Toast.LENGTH_LONG).show();
                }
                catch (Exception e)
                {

Toast.makeText(getBaseContext(),e.getMessage(),Toast.LENGTH_LONG).show();
                }
            }
        });

        read.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v)
            {
                String message;
                String buf = "";
                try
                {
                    File f = new File("/sdcard/myfile.txt");
                    FileInputStream fin = new FileInputStream(f);
                    BufferedReader br = new BufferedReader(new
InputStreamReader(fin));
                    while ((message = br.readLine()) != null)
                    {
                        buf += message;
                    }
                    e1.setText(buf);
                    br.close();
                    fin.close();
                    Toast.makeText(getBaseContext(),"Data Recived from
SDCARD",Toast.LENGTH_LONG).show();
                }
                catch (Exception e)
                {
                    Toast.makeText(getBaseContext(), e.getMessage(),
Toast.LENGTH_LONG).show();
                }
            }
        });

        clear.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v)
            {
                e1.setText("");
            }
        });
    }

}
```

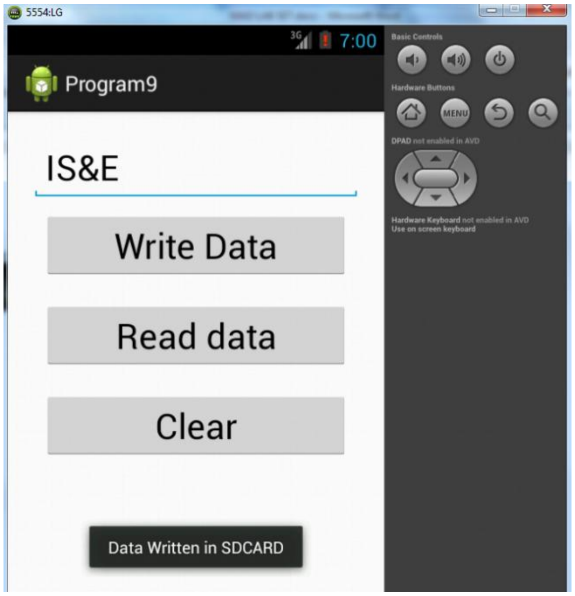
- So now the Coding part of Program 9 is also completed.
- Now run the application to see the output.

Output:



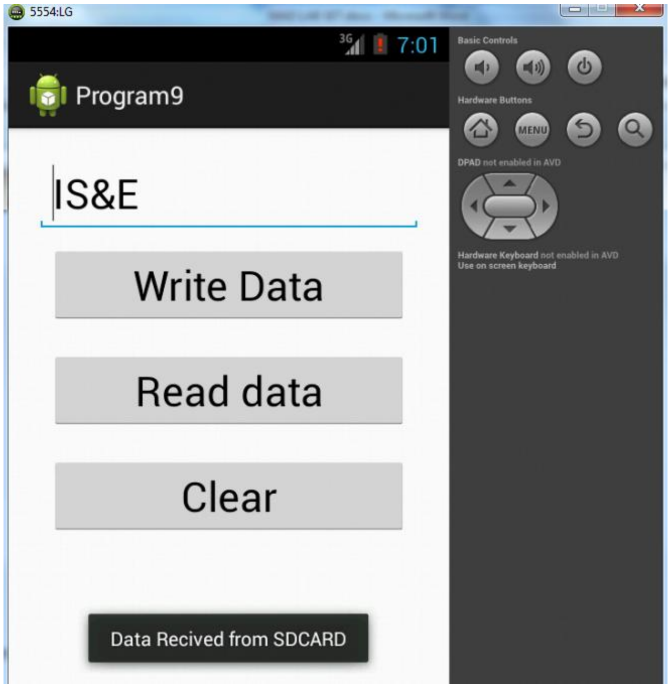


IS&E



IS&E

Data Written in SDCARD



IS&E

Data Recived from SDCARD

Result:

Thus Android Application that writes data to the SD Card is developed and executed successfully.

<u>PROGRAM 7</u>

<u>Aim:</u>

To build an application that implements multi-threading concept.

<u>Procedure:</u>

Create Main Activity for the Android Application

Designing layout for the Android Application:

Designing Layout for Main Activity:

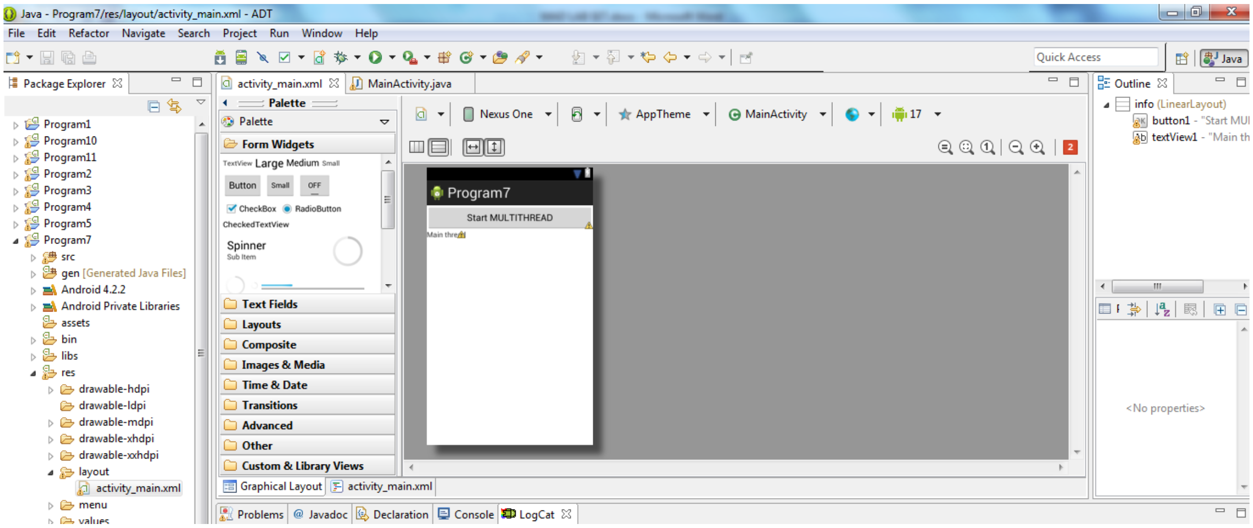
- Click on **Program7 -> res -> layout -> activity_main.xml.**

- Now go to XML code editor.
- Then delete the code which is there and type the code as given below.

Code for 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:id="@+id/info" >
      <Button
        android:id="@+id/button1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="fetchData"
        android:text="Start MULTITHREAD" />
    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Main thread" />
</LinearLayout>
```

- Now click on **Design** and your application will look as given below.



- So now the designing part is completed.

Java Coding for the Android Application:

Java Coding for Main Activity:

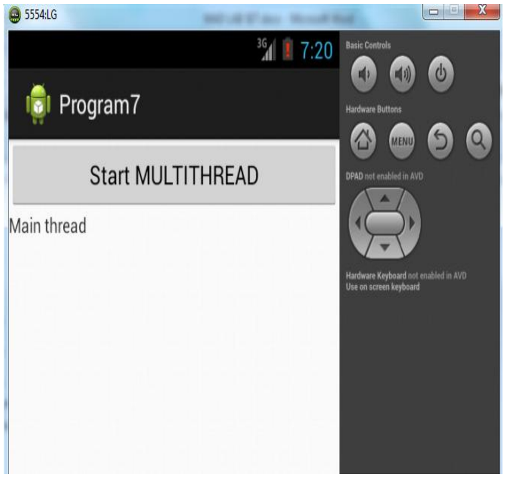
- Click on **Program7 -> java -> com.example.Program7 -> MainActivity.**

Code for MainActivity.java

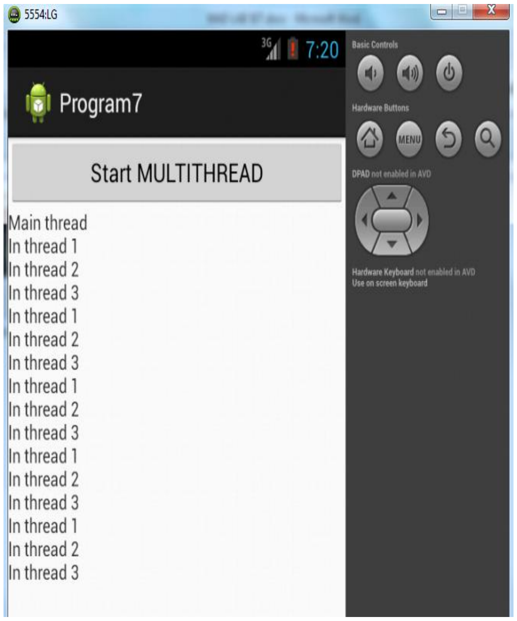
```
package com.example.program7;

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


import android.os.Handler;
import android.view.View;
import android.widget.TextView;


public class MainActivity extends Activity {

        private  TextView tvOutput;
          private  static final int t1 = 1;
          private  static final int t2 = 2;
          private  static final int t3 = 3;
          @Override
          public void onCreate(Bundle savedInstanceState) {
              super.onCreate(savedInstanceState);
              setContentView(R.layout.activity_main);
              tvOutput = (TextView) findViewById(R.id.textView1);

          }
          public void fetchData(View v) {
              tvOutput.setText("Main thread");
              thread1.start();
              thread2.start();
              thread3.start();
          }

          Thread thread1 = new Thread(new Runnable() {

          @Override
          public void run() {
          for (int i = 0; i < 5; i++) {
          try {
          Thread.sleep(1000);
          } catch (InterruptedException e) {
          e.printStackTrace();
          }
          handler.sendEmptyMessage(t1);
          }

          }
          });
          Thread thread2 = new Thread(new Runnable() {

              @Override
              public void run() {
              for (int i = 0; i < 5; i++) {
              try {
              Thread.sleep(1000);
              } catch (InterruptedException e) {
              e.printStackTrace();
              }
              handler.sendEmptyMessage(t2);
              }

              }
              });
          Thread thread3 = new Thread(new Runnable() {

              @Override
              public void run() {
              for (int i = 0; i < 5; i++) {
              try {
              Thread.sleep(1000);
              } catch (InterruptedException e) {
              e.printStackTrace();
              }
              handler.sendEmptyMessage(t3);
              }

              }
              });

          Handler handler = new Handler() {
          public void handleMessage(android.os.Message msg) {
          if(msg.what == t1) {
          tvOutput.append("\nIn thread 1");
          }
          if(msg.what == t2) {
              tvOutput.append("\nIn thread 2");
              }
          if(msg.what == t3) {
              tvOutput.append("\nIn thread 3");
              }
          }
          };


}
```

- So now the Coding part of Program 7 is also completed.
- Now run the application to see the output.

Output:



```
Main thread
```



```
Main thread
In thread 1
In thread 2
In thread 3
In thread 1
In thread 2
In thread 3
In thread 1
In thread 2
In thread 3
In thread 1
In thread 2
In thread 3
In thread 1
In thread 2
In thread 3
```

Result:

Thus Android Application that implements multi-threading concept is developed and executed successfully.

<u>PROGRAM 8</u>

<u>Aim:</u>

To develop a native application that uses GPS location information.

<u>Procedure:</u>

Create Main Activity for the Android Application

Designing layout for the Android Application:

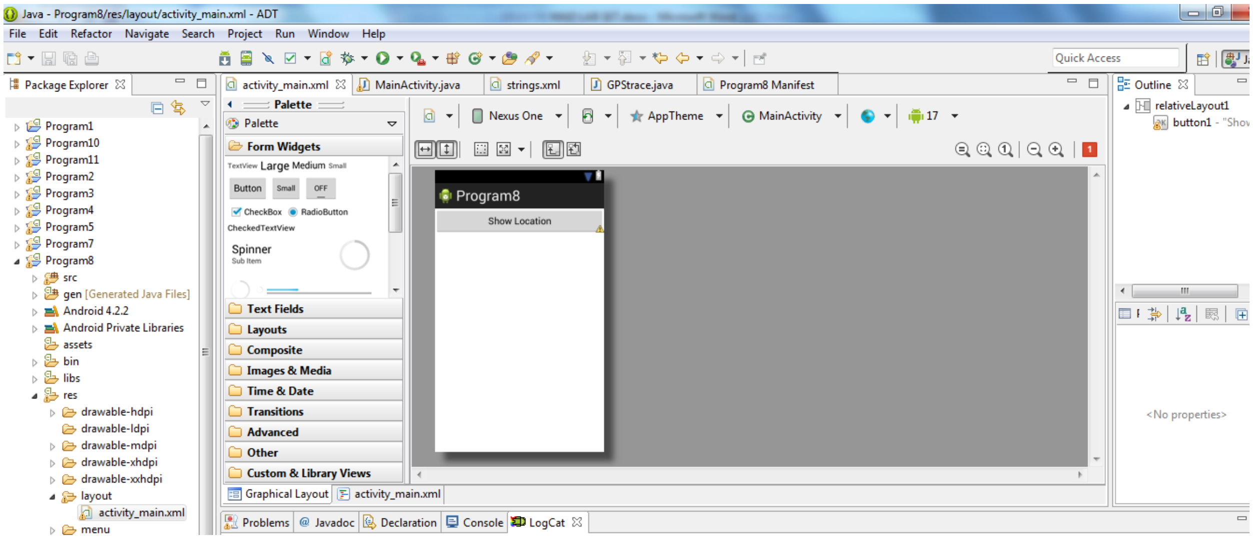
Designing Layout for Main Activity:

- Click on **Program8 -> res -> layout -> activity_main.xml.**
- Now go to XML code editor.
- Then delete the code which is there and type the code as given below.

Code for Activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
    <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:id="@+id/relativeLayout1"
        android:layout_width="match_parent"
        android:layout_height="match_parent" >
 <Button
        android:id="@+id/button1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Show Location"
        />

</RelativeLayout>
```

- Now click on **Design** and your application will look as given below.



- So now the designing part is completed.

Changes in Manifest for the Android Application:

- Click on **Program8 -> manifests -> AndroidManifest.xml**
  - Now go to XML code editor.
  - Then delete the code which is there and type the code as given below.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.program8"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />

    <uses-permission
    android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission
    android:name="android.permission.INTERNET"/>


    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.example.program8.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>
    </application>

</manifest>
```

Java Coding for the Android Application:

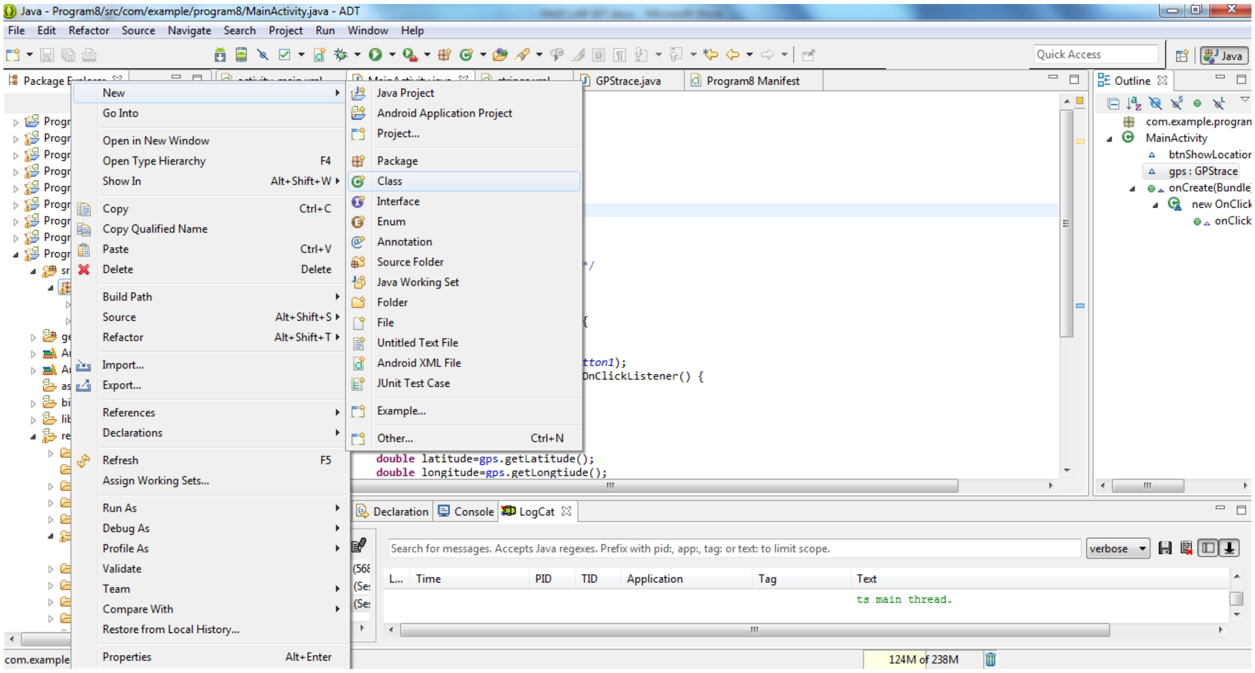
Java Coding for Main Activity:

- Click on **Program8 -> java -> com.example.Program8 -> MainActivity.**

Code for MainActivity.java

```
package com.example.program8;

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


public class MainActivity extends Activity {

        /** Called when the activity is first created. */
    Button btnShowLocation;
    GPStrace gps;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnShowLocation=(Button)findViewById(R.id.button1);
        btnShowLocation.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                gps=new GPStrace(MainActivity.this);
                if(gps.canGetLocation()){
                    double latitude=gps.getLatitude();
                    double longitude=gps.getLongtiude();
                    Toast.makeText(getApplicationContext(),"Your Location is 
\nLat:"+latitude+"\nLong:"+longitude, Toast.LENGTH_LONG).show();
                }
                else
                {
                    gps.showSettingAlert();
                            }
            }
        });
    }


}
```

Create one more Java class file Named as “GPStrace.java” in the same package as shown below



```
*/
{
tton1);
OnClickListener() {
double latitude=gps.getLatitude();
double longitude=gps.getLongtiude();
```

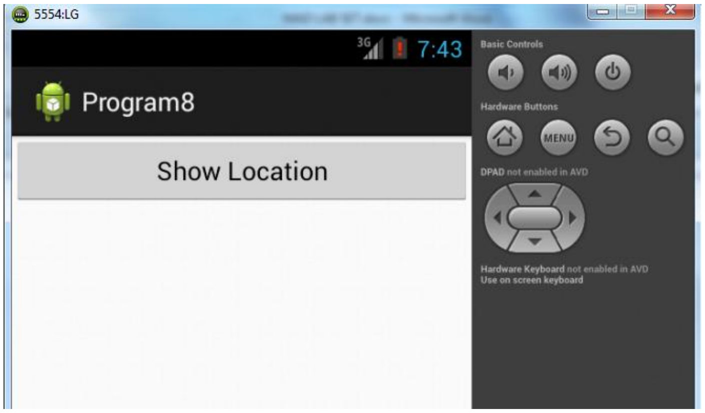
```
ts main thread.
```

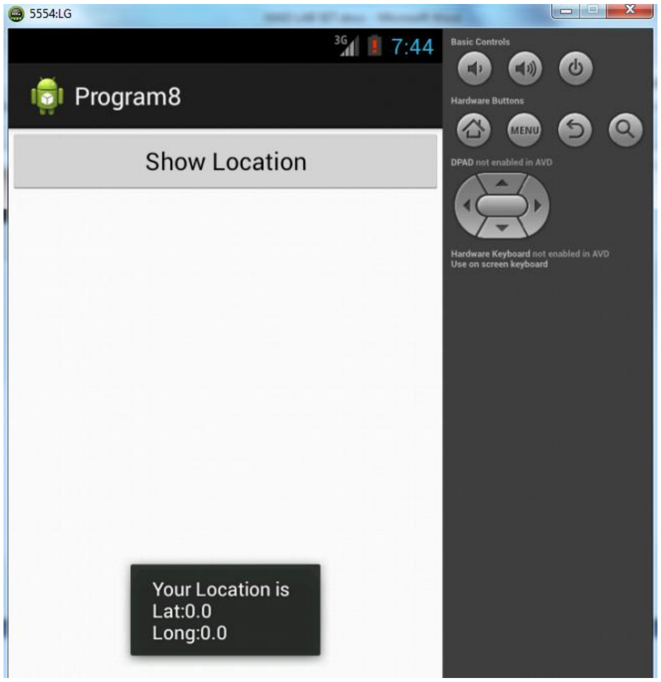
Code for GPStrace.java

```
package com.example.program8;

import android.app.AlertDialog;
import android.app.Service;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.os.IBinder;
import android.provider.Settings;

public class GPStrace extends Service implements LocationListener{
        private final Context context;
        boolean isGPSEnabled=false;
        boolean canGetLocation=false;
        boolean isNetworkEnabled=false;
        Location location;
        double latitude;
        double longtitude;
        private static final long MIN_DISTANCE_CHANGE_FOR_UPDATES=10;
        private static final long MIN_TIME_BW_UPDATES=1000*60*1;
        protected LocationManager locationManager;
        public GPStrace(Context context)
        {
            this.context=context;
            getLocation();
        }
        public Location getLocation()
        {
            try{
                locationManager=(LocationManager)
context.getSystemService(LOCATION_SERVICE);

isGPSEnabled=locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER);

isNetworkEnabled=locationManager.isProviderEnabled(LocationManager.NETWORK_PROVIDER);
                if(!isGPSEnabled && !isNetworkEnabled){

                }else{
                    this.canGetLocation=true;
                    if(isNetworkEnabled){

                        locationManager.requestLocationUpdates(
                            LocationManager.NETWORK_PROVIDER,
                            MIN_TIME_BW_UPDATES,
                            MIN_DISTANCE_CHANGE_FOR_UPDATES,this);

                        }
                        if(locationManager!=null){

location=locationManager.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
                            if(location !=null){
                                latitude=location.getLatitude();
                                longtitude=location.getLongitude();

                            }
                        }
                    }
                    if(isGPSEnabled){
                        if(location==null){

locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,MIN_TIME_BW_UPDAT
ES, MIN_DISTANCE_CHANGE_FOR_UPDATES, this);
                            if(locationManager!=null){

location=locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
                            if(location!=null){
                                latitude=location.getLatitude();
                                longtitude=location.getLongitude();
                            }
                            }
                        }
                    }
                }

            catch(Exception e)
            {
                e.printStackTrace();
            }
            return location;
        }
        public void stopUsingGPS(){
            if(locationManager!=null){
                locationManager.removeUpdates(GPStrace.this);
            }
        }
        public double getLatitude(){
            if(location!=null){
                latitude=location.getLatitude();
            }
            return latitude;
        }
        public double getLongtiude(){
            if(location!=null){
                longtitude=location.getLatitude();
            }
            return longtitude;
        }
        public boolean canGetLocation(){
            return this.canGetLocation;
        }
        public void showSettingAlert(){
            AlertDialog.Builder alertDialog=new AlertDialog.Builder(context);
            alertDialog.setTitle("GPS is settings");
            alertDialog.setMessage("GPS is not enabled.Do you want to go to setting 
menu?");
            alertDialog.setPositiveButton("settings", new 
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog,int which){
                    Intent intent=new 
Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
                    context.startActivity(intent);
                }
            });
            alertDialog.setNegativeButton("cancel", new 
DialogInterface.OnClickListener() {


                @Override
                public void onClick(DialogInterface dialog, int which) {
                    // TODO Auto-generated method stub
                    dialog.cancel();
                }
        });
        alertDialog.show();
        }
    @Override
    public void onLocationChanged(Location location) {
        // TODO Auto-generated method stub

    }
    @Override
    public void onProviderDisabled(String provider) {
        // TODO Auto-generated method stub

    }
    @Override
    public void onProviderEnabled(String provider) {
        // TODO Auto-generated method stub

    }
    @Override
    public void onStatusChanged(String provider, int status, Bundle extras) {
        // TODO Auto-generated method stub

    }
    @Override
    public IBinder onBind(Intent intent) {
        // TODO Auto-generated method stub
        return null;
    }

}
```

- So now the Coding part of Program 8 is also completed.
- Now run the application to see the output.

Output:





```
Your Location is
Lat:0.0
Long:0.0
```

Result:

Thus Android Application that uses GPS location information is developed and executed successfully.

http://www.codingconnect.net/android-application-for-layout-managers-event-listners/