SAVEETHA
SCHOOL OF ENGINEERING

ITA0352
Mobile Computing for Utility Applications

Record Note Book

Name:_____

Register No:_____

Department:_____

SAVEETHA
INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES
Saveetha Nagar, Thandalam, Chennai - 602 105.
Tamil Nadu, India. Phone : +91 44 6672 6672
Website : www.saveetha.com,
Email : principalsse@saveetha.com

SAVEETHA
SCHOOL OF ENGINEERING

Department Of .....

LABORATORY RECORD NOTE BOOK

20 - 20

This is certify that this is a bonafide record of that work done by

Mr. / Ms .....Register Number .....

of the year ..... B.E / B.Tech., Department of .....

in the ..... Laboratory in the ..... Semester

University Examination held on .....

Staff in - Charge

Head of the Department

Internal Examiner

External Examiner

SAVEETHA
INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES
(Declared as Deemed to be University under Section 3 of UGC Act 1956)

Saveetha Nagar, Thandalam, Chennai - 602 105,
Tamil Nadu, India. Phone : +91 44 6672 6672
Website : www.saveetha.com,
Email : principal.sse@saveetha.com

TABLE OF CONTENTS

Exp: No: 1	Develop an application that uses GUI components, Font and Colours

Aim:

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

Procedure:

1. Start the process.

2. Start the Eclipse IDE for developing mobile application.

3. Create a new project by : File ->New -> Project -> Android (Wizard) - > Android Application Project .

4. In New Android Application dialog to enter the required values for creating the new android application such as Application Name, Package name and Project Name.

5. Enter the Application name and then click Next.

6. In Under create activity wizard Add an activity to <*template*>, select Blank Activity and click Next.

7. Change the Activity Name to *Mainactivity* then Click the Finish button to create the project.

8. Android Project is created which contains the following file

simpledraw/res/layout/activity_mainactivity.xml

This is the XML layout file for the activity you added when you created the project with Android Studio.

simpledraw/src/com.exmaple.simpledraw/ Mainactivity.java

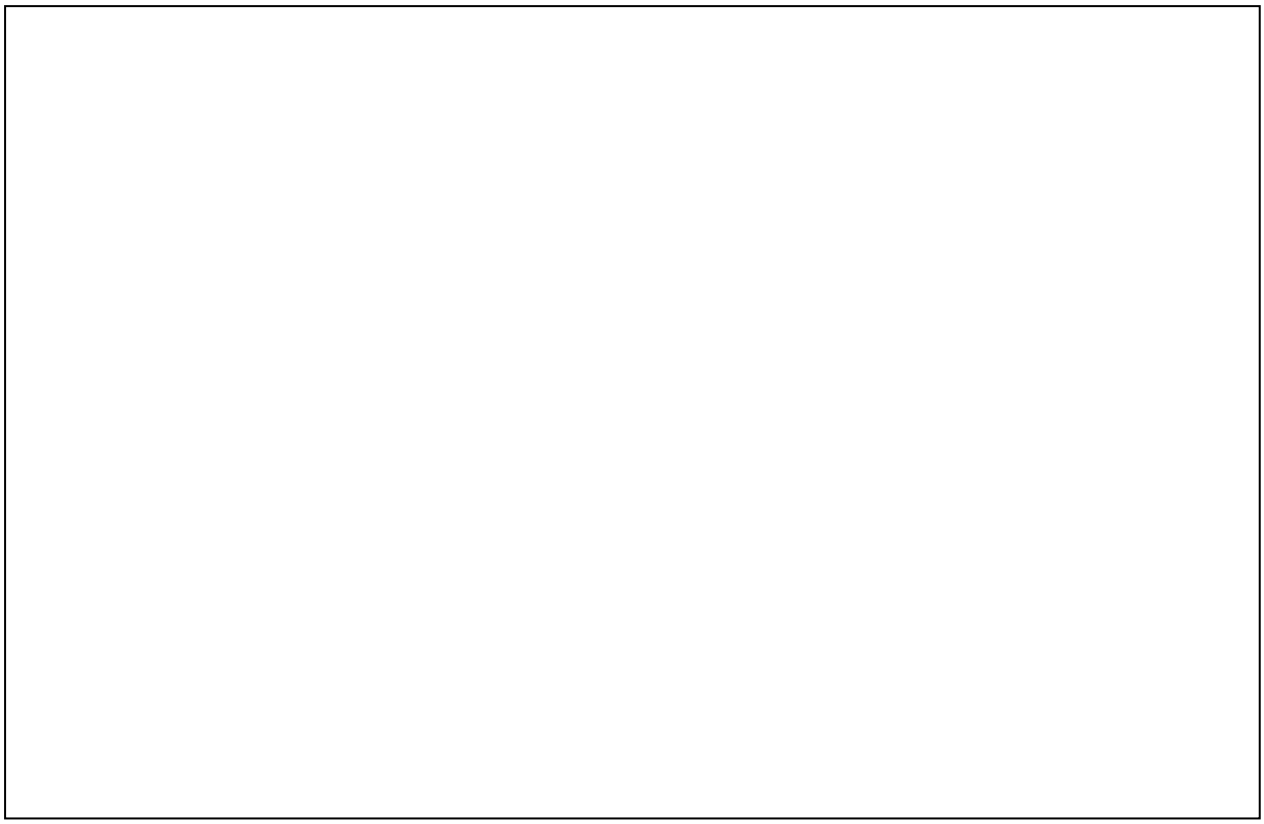
When you select the file you see the class definition for the activity you created.

simpledraw/AndroidManifest.xml

Steps for Creating Emulator And Run :

To run your app on the emulator you need to first create an Android Virtual Device (AVD). An AVD is a device configuration for the Android emulator that allows you to model a specific device.

Graphical Layout



Code for Login

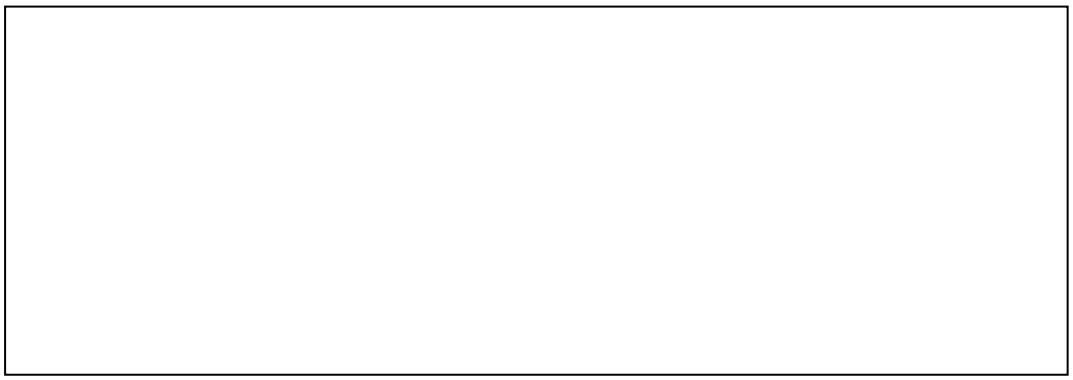
```
package com.example.firstproject;
import android.app.Activity;
import android.graphics.Color;
import android.graphics.Typeface;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.TextView;
public class MainActivity extends Activity {
float font =20;
```

```
int count = 1;
Button b1,b2,b3;
@Override
public void onCreate(Bundle savedInstanceState){
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
final TextView t1 = (TextView)
findViewById(R.id.textView1);
b1 = (Button) findViewById(R.id.button1);
b1.setOnClickListener(new View.OnClickListener(){
public void onClick(View view){
t1.setText("WELCOME TO ANDROID");
t1.setTextSize(font);
font=font+5;
if (font == 50)
font = 20;
}
});
b2 = (Button) findViewById(R.id.button2);
b2.setOnClickListener(new View.OnClickListener() {
public void onClick(View view) {
t1.setText("WELCOME TO ANDROID");
switch(count){
case 1:
t1.setTextColor(Color.parseColor("#7f00ff"));
break;
case 2:
```

```
t1.setTextColor(Color.parseColor("#00FF00"));
break;
case 3:
t1.setTextColor(Color.parseColor("#FF0000"));
break;
case 4:
t1.setTextColor(Color.parseColor("#0000FF"));
break;
}
count++;
if (count == 5)
count = 1;
}
});
b3 = (Button) findViewById(R.id.button3);
b3.setOnClickListener(new OnClickListener() {
@Override
public void onClick(View view){
t1.setText("WELCOME TO ANDROID");
switch (count){
case 1:
t1.setTypeface(Typeface.DEFAULT,
Typeface.ITALIC);
break;
case 2:
t1.setTypeface(Typeface.MONOSPACE,
Typeface.NORMAL);
```

```
break;
case 3:
t1.setTypeface(Typeface.SANS_SERIF,
Typeface.BOLD);
break;
case 4:
t1.setTypeface(Typeface.SERIF,
Typeface.BOLD_ITALIC);
break;
}
count++;
if (count == 5)
count = 1;
}
});
}
}
```

OUTPUT



RESULT:

Exp: No: 2	Develop an application that uses Layout Managers and event listeners.

Aim:

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

Procedure:

1. Start the process.

2. Start the Eclipse IDE for developing mobile application.

3. Create a new project by : File ->New -> Project -> Android (Wizard) - > Android Application Project .

4. In New Android Application dialog to enter the required values for creating the new android application such as Application Name, Package name and Project Name.

5. Enter the Application name and then click Next.

6. In Under create activity wizard Add an activity to <*template*>, select Blank Activity and click Next.

7. Change the Activity Name to *Mainactivity* then Click the Finish button to create the project.

8. Android Project is created which contains the following file

simpledraw/res/layout/activity_mainactivity.xml

This is the XML layout file for the activity you added when you created the project with Android Studio.

simpledraw/src/com.exmaple.simpledraw/ Mainactivity.java

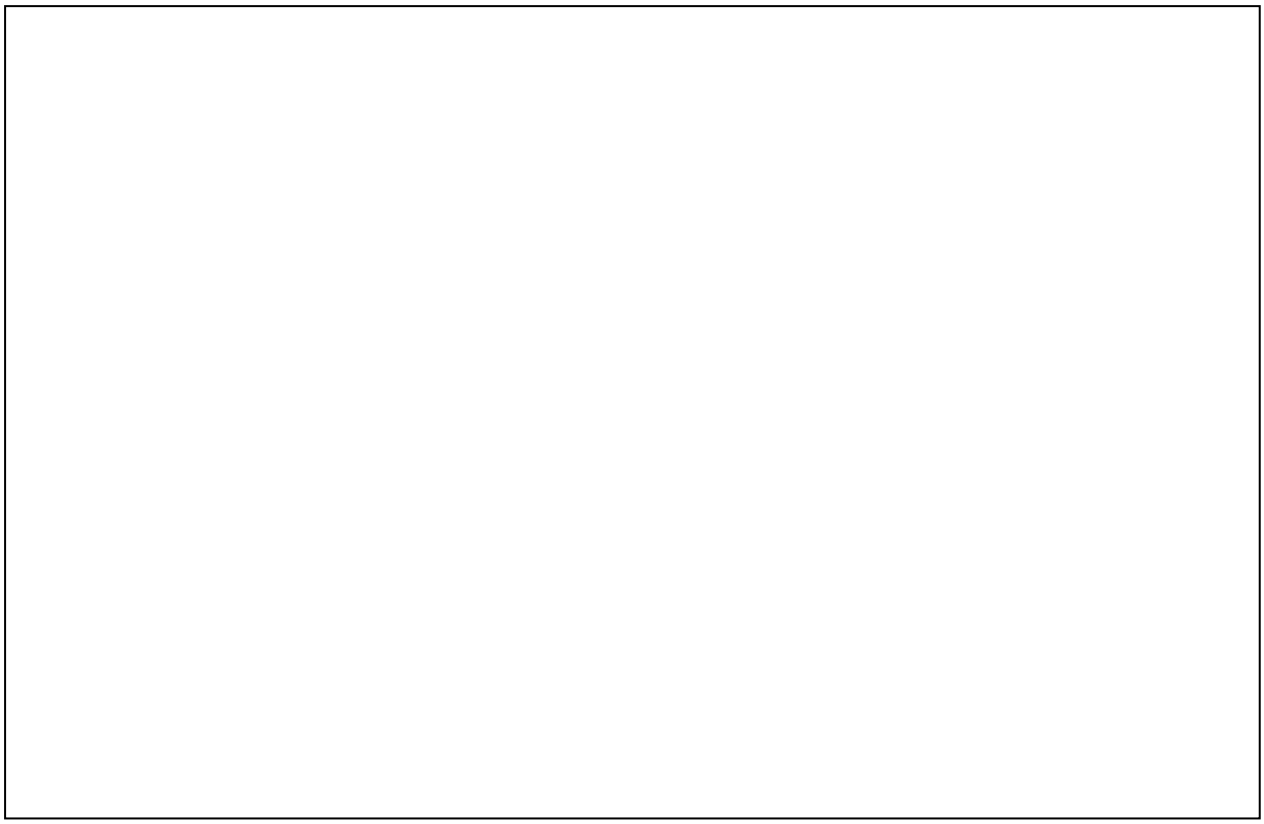
When you select the file you see the class definition for the activity you created.

simpledraw/AndroidManifest.xml

Steps for Creating Emulator And Run :

To run your app on the emulator you need to first create an Android Virtual Device (AVD). An AVD is a device configuration for the Android emulator that allows you to model a specific device.

Graphical Layout



Code for Login

```
package com.example.app2;

import android.app.Activity;

import android.os.Bundle;

import android.view.Menu;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.AdapterView;

import android.widget.AdapterView.OnItemClickListener;

import android.widget.AdapterView.OnItemSelectedListener;

import android.widget.Button;

import android.widget.EditText;
```

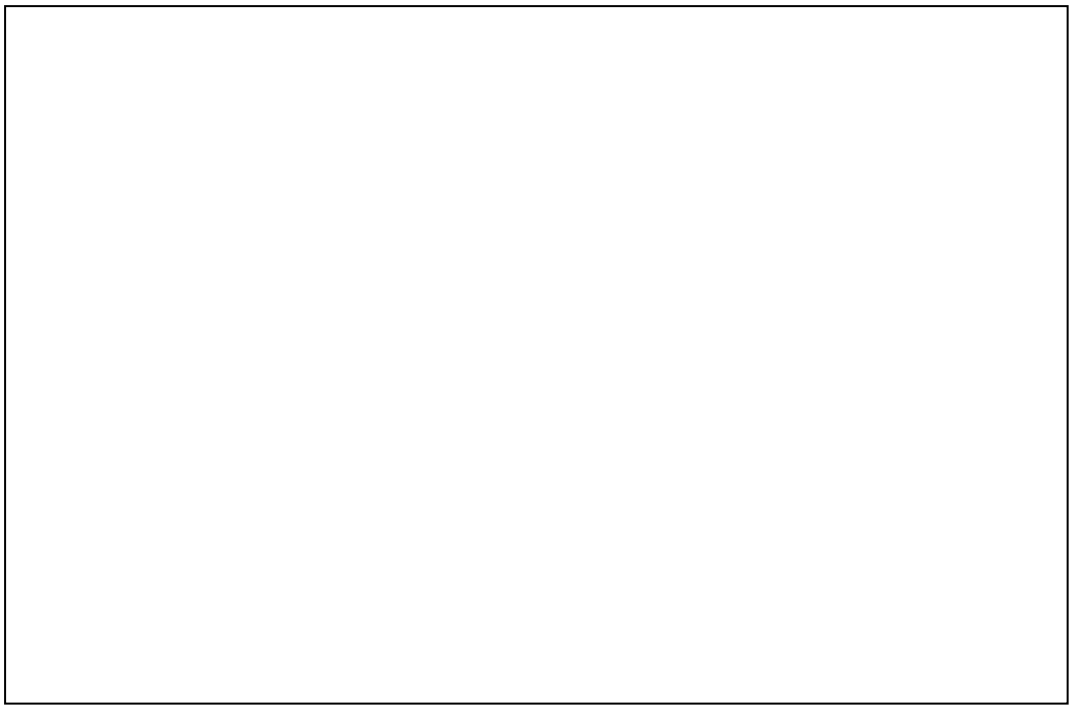
```
import android.widget.ListView;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.RadioGroup.OnCheckedChangeListener;
import android.widget.RatingBar;
import android.widget.RatingBar.OnRatingBarChangeListener;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;
public class MainActivity extends Activity {
EditText et;
Spinner sp;
RadioGroup rg;
RadioButton r1, r2;
ListView lv;
RatingBar rb;
Button b;
String name;
int degree;
String area;
String rating;
String sex;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
et = (EditText) findViewById(R.id.editText1);
```

```
sp = (Spinner) findViewById(R.id.spinner1);
rg = (RadioGroup) findViewById(R.id.radioGroup1);
r1 = (RadioButton) findViewById(R.id.radioButton1);
r2 = (RadioButton) findViewById(R.id.radioButton2);
lv = (ListView) findViewById(R.id.listView1);
rb = (RatingBar) findViewById(R.id.ratingBar1);
b = (Button) findViewById(R.id.button1);
// Spinner Item
sp.setOnItemSelectedListener(new OnItemSelectedListener() {
@Override
public void onItemSelected(AdapterView<?> arg0, View arg1, int arg2, long arg3) {
sex = arg0.getItemAtPosition(arg2).toString();
}
@Override
public void onNothingSelected(AdapterView<?> arg0) {
}
});
// RadioGroup Item
rg.setOnCheckedChangeListener(new OnCheckedChangeListener() {
@Override
public void onCheckedChanged(RadioGroup arg0, int arg1) {
String temp;
View radioButton = arg0.findViewById(arg1);
degree = arg0.indexOfChild(radioButton);
if (degree == 0)
temp = "B.E CSE";
else
```

```
temp = "B.TECH IT";
}
});
lv.setOnItemClickListener(new AdapterView.OnItemClickListener() {
@Override
public void onItemClick(AdapterView<?> arg0, View arg1, int arg2, long arg3) {
area = ((TextView) arg1).getText().toString();
}
});
rb.setOnRatingBarChangeListener(new RatingBar.OnRatingBarChangeListener() {
@Override
public void onRatingChanged(RatingBar arg0, float arg1, boolean arg2) {
rating = String.valueOf(arg1);
}
});
b.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View arg0) {
name = et.getText().toString();
String line = System.getProperty("line.separator");
String str = "Name = " + name + line + "Sex = " + sex + "Degree = " + degree + line +
"Area = " + area + line + "Rating = " + rating;
Toast.makeText(MainActivity.this, str, Toast.LENGTH_LONG).show();
Toast.makeText(MainActivity.this, "Your Data Saved!!!",
Toast.LENGTH_LONG).show();
}
});
```

```
}
@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;
}
}
```

OUTPUT



RESULT:

Exp: No: 3	Develop an application that draws basic graphical primitives on the screen

Aim:

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

Procedure:

1. Start the process.

2. Start the Eclipse IDE for developing mobile application.

3. Create a new project by : File ->New -> Project -> Android (Wizard) - > Android Application Project .

4. In New Android Application dialog to enter the required values for creating the new android application such as Application Name, Package name and Project Name.

5. Enter the Application name and then click Next.

6. In Under create activity wizard Add an activity to <*template*>, select Blank Activity and click Next.

7. Change the Activity Name to *Mainactivity* then Click the Finish button to create the project.

8. Android Project is created which contains the following file

simpledraw/res/layout/activity_mainactivity.xml

This is the XML layout file for the activity you added when you created the project with Android Studio.

simpledraw/src/com.exmaple.simpledraw/ Mainactivity.java

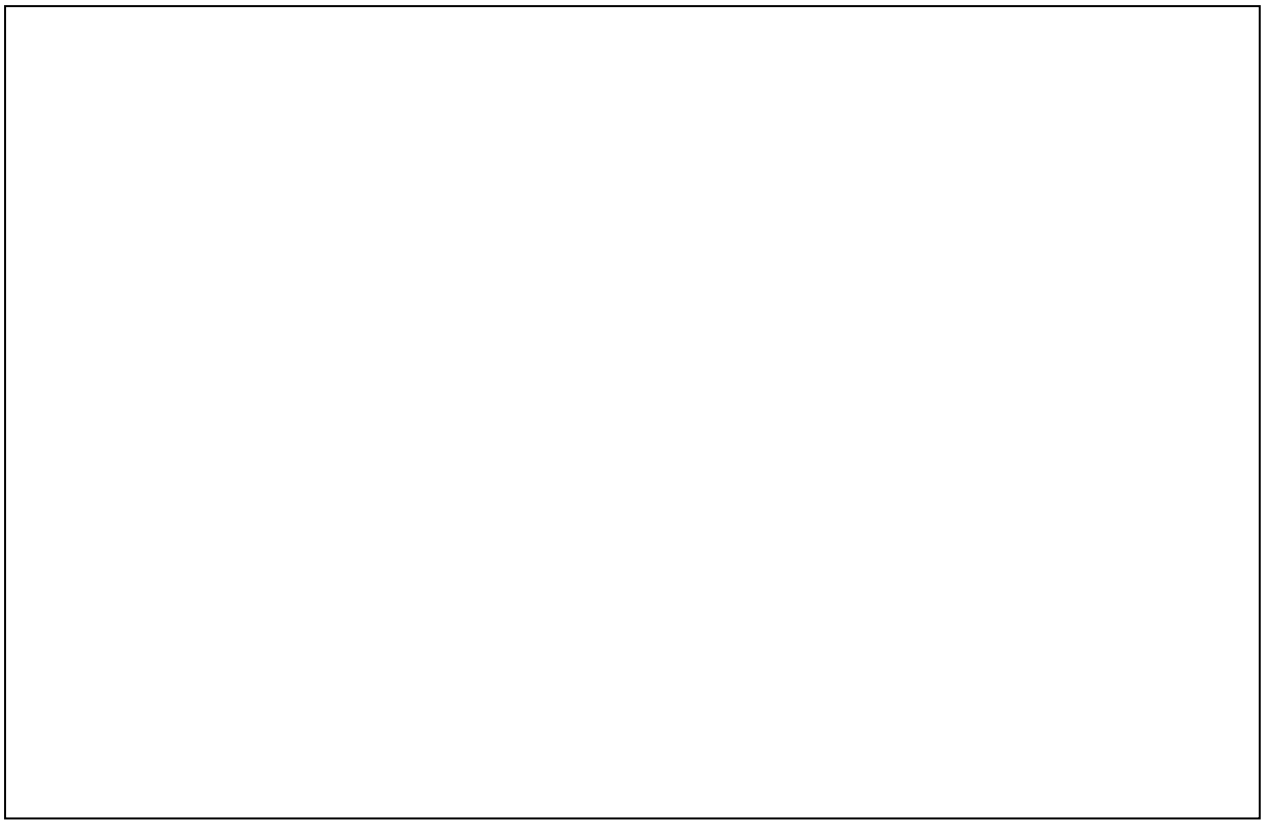
When you select the file you see the class definition for the activity you created.

simpledraw/AndroidManifest.xml

Steps for Creating Emulator And Run :

To run your app on the emulator you need to first create an Android Virtual Device (AVD). An AVD is a device configuration for the Android emulator that allows you to model a specific device.

Graphical Layout



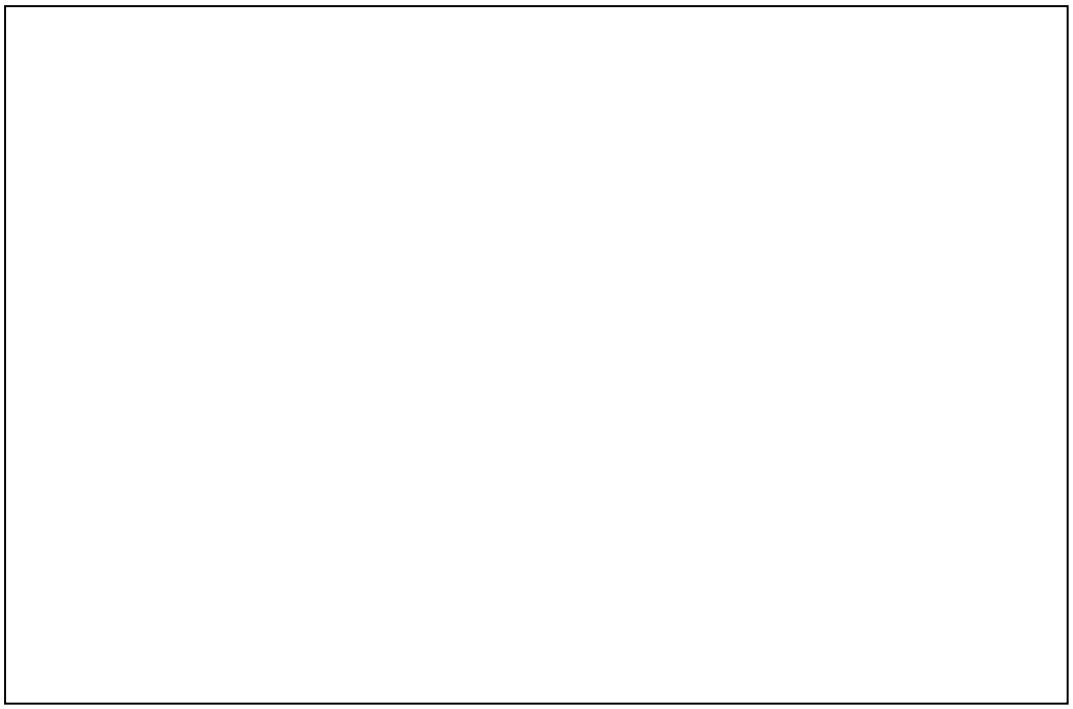
Code for Login

```
package com.example.app5graphics;
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.os.Bundle;
import android.widget.ImageView;
public class MainActivity extends Activity
{
```

```
@Override
@SuppressWarnings("deprecation")
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);

}

}
```

OUTPUT



RESULT:

Exp: No: 4	Develop an application that makes use of databases

Aim:

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

Procedure:

1. Start the process.

2. Start the Eclipse IDE for developing mobile application.

3. Create a new project by : File ->New -> Project -> Android (Wizard) - > Android Application Project .

4. In New Android Application dialog to enter the required values for creating the new android application such as Application Name, Package name and Project Name.

5. Enter the Application name and then click Next.

6. In Under create activity wizard Add an activity to <*template*>, select Blank Activity and click Next.

7. Change the Activity Name to *Mainactivity* then Click the Finish button to create the project.

8. Android Project is created which contains the following file

simpledraw/res/layout/activity_mainactivity.xml

This is the XML layout file for the activity you added when you created the project with Android Studio.

simpledraw/src/com.exmaple.simpledraw/ Mainactivity.java

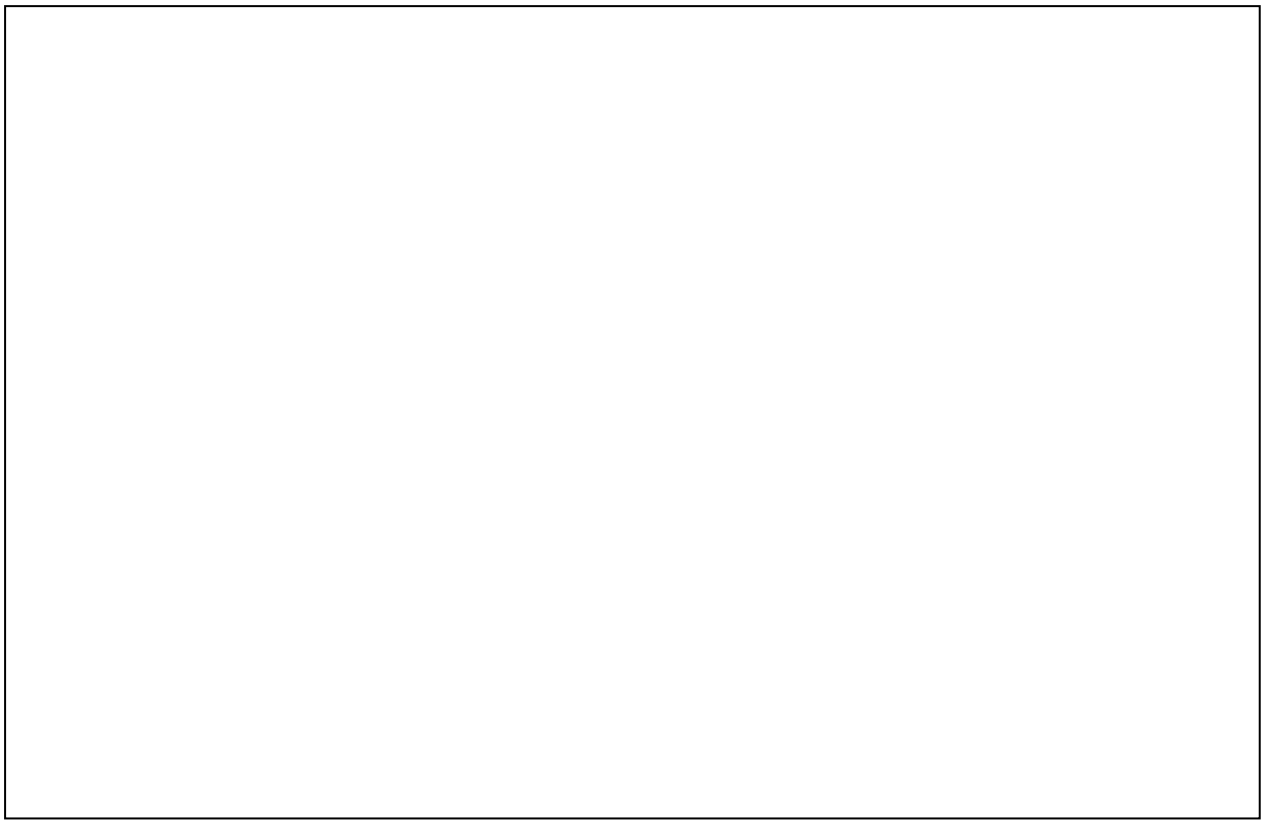
When you select the file you see the class definition for the activity you created.

simpledraw/AndroidManifest.xml

Steps for Creating Emulator And Run :

To run your app on the emulator you need to first create an Android Virtual Device (AVD). An AVD is a device configuration for the Android emulator that allows you to model a specific device.

Graphical Layout



Code for Login

```
package com.example.exp7;

import android.app.Activity;

import android.app.AlertDialog.Builder;

import android.content.Context;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.os.Bundle;

import android.view.View;

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.editText2);
Name=(EditText)findViewById(R.id.editText1);
Marks=(EditText)findViewById(R.id.editText3);
Insert=(Button)findViewById(R.id.button1);
Delete=(Button)findViewById(R.id.button2);
Update=(Button)findViewById(R.id.button3);
View=(Button)findViewById(R.id.button4);
ViewAll=(Button)findViewById(R.id.button5);
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(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)
```

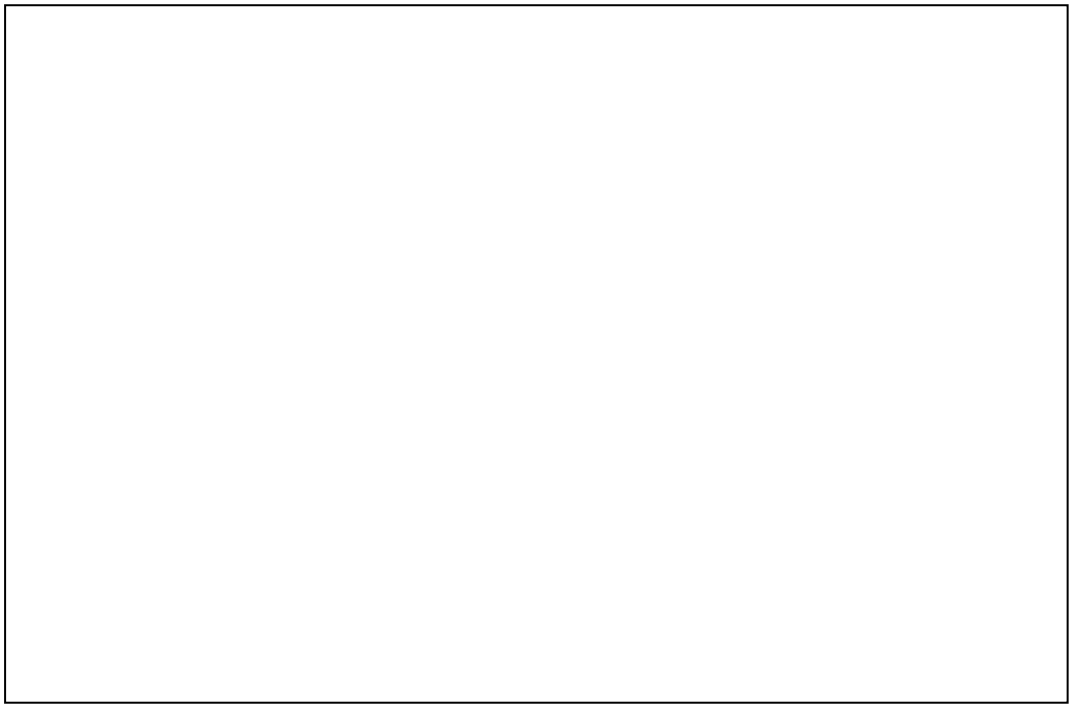
```
{
showMessage("Error", "Please enter Rollno");
return;
}
Cursor c=db.rawQuery("SELECT * FROM student WHERE rollno='"+Rollno.getText()+"'", null);
if(c.moveToFirst())
{
db.execSQL("DELETE FROM student WHERE rollno='"+Rollno.getText()+"'");
showMessage("Success", "Record Deleted");
}
else
{
showMessage("Error", "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("");
}
{
}
}
```

OUTPUT



RESULT:

Exp: No: 5	Implement an application that uses Multi-threading

Aim:

To develop a Android Application that implements Multi threading.

Procedure:

1. Start the process.

2. Start the Eclipse IDE for developing mobile application.

3. Create a new project by : File ->New -> Project -> Android (Wizard) - > Android Application Project .

4. In New Android Application dialog to enter the required values for creating the new android application such as Application Name, Package name and Project Name.

5. Enter the Application name and then click Next.

6. In Under create activity wizard Add an activity to <*template*>, select Blank Activity and click Next.

7. Change the Activity Name to *Mainactivity* then Click the Finish button to create the project.

8. Android Project is created which contains the following file

simpledraw/res/layout/activity_mainactivity.xml

This is the XML layout file for the activity you added when you created the project with Android Studio.

simpledraw/src/com.exmaple.simpledraw/ Mainactivity.java

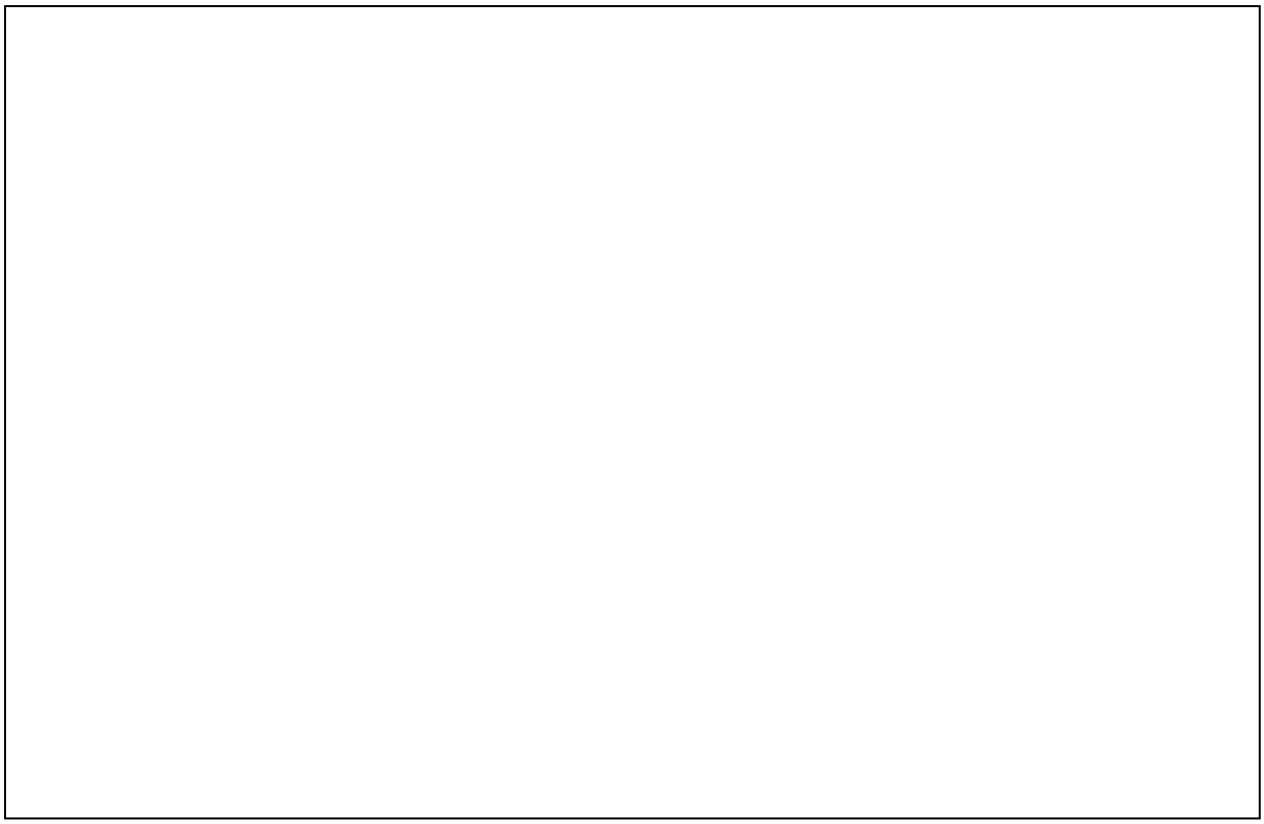
When you select the file you see the class definition for the activity you created.

simpledraw/AndroidManifest.xml

Steps for Creating Emulator And Run :

To run your app on the emulator you need to first create an Android Virtual Device (AVD). An AVD is a device configuration for the Android emulator that allows you to model a specific device.

Graphical Layout



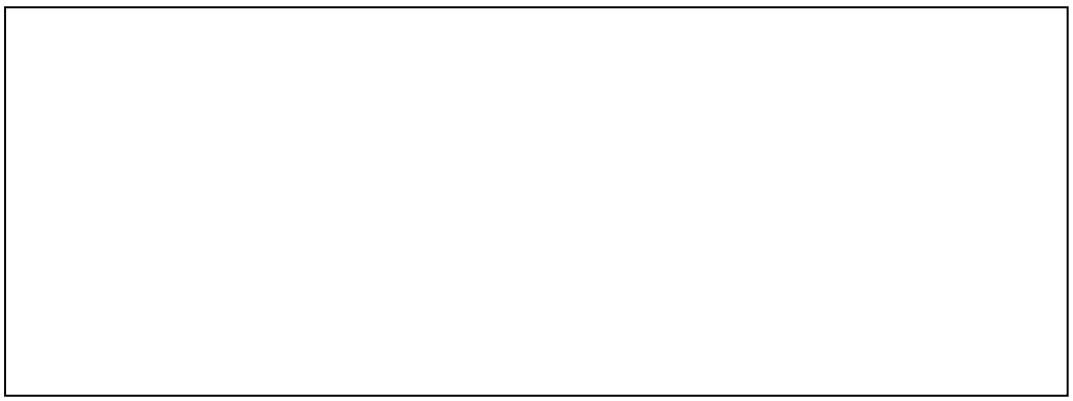
Code for Login

```
package com.example.app4mt;

import android.app.Activity;

import android.os.Bundle;

import android.view.Menu;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.ProgressBar;

import android.widget.TextView;

import android.widget.Toast;

public class MainActivity extends Activity {
```

```
ProgressBar pb;
TextView tv;
Button b;
int i;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
pb = (ProgressBar) findViewById(R.id.progressBar1);
tv = (TextView) findViewById(R.id.textView1);
b = (Button) findViewById(R.id.button1);
b.setOnClickListener(new OnClickListener() {
@Override
public void onClick(View arg0) {
Runnable r = new Runnable() {
@Override
public void run() {
for (i = 20; i <= 100; i += 20) {
final int value = i;
try {
Thread.sleep(2000);
} catch (Exception e) {}
pb.post(new Runnable() {
@Override
public void run() {
tv.setText(String.valueOf(value) + "%");
pb.setProgress(value);
```

```
if (value == 100)
Toast.makeText(MainActivity.this, "Download Completed...",3000).show();
}
});
}
}
};
new Thread(r).start();
}
});
}
@Override
public boolean onCreateOptionsMenu(Menu menu) {
getMenuInflater().inflate(R.menu.main, menu);
return true;
}
}
```

OUTPUT



RESULT:

Exp: No: 6	Develop a Native Calculator Application

Aim:

To develop a Simple Android Application for Native Calculator.

Procedure:

1. Start the process.

2. Start the Eclipse IDE for developing mobile application.

3. Create a new project by : File ->New -> Project -> Android (Wizard) - > Android Application Project .

4. In New Android Application dialog to enter the required values for creating the new android application such as Application Name, Package name and Project Name.

5. Enter the Application name and then click Next.

6. In Under create activity wizard Add an activity to <*template*>, select Blank Activity and click Next.

7. Change the Activity Name to *Mainactivity* then Click the Finish button to create the project.

8. Android Project is created which contains the following file

simpledraw/res/layout/activity_mainactivity.xml

This is the XML layout file for the activity you added when you created the project with Android Studio.

simpledraw/src/com.exmaple.simpledraw/ Mainactivity.java

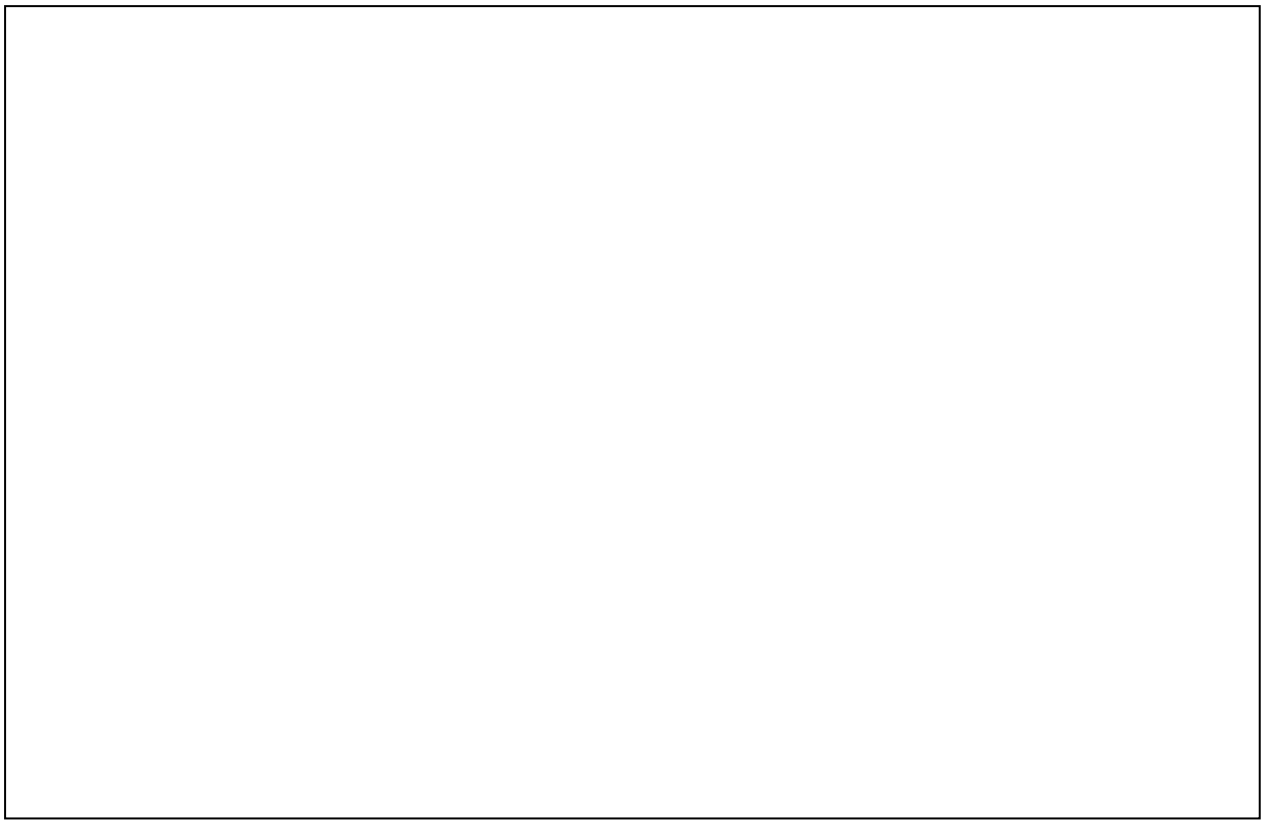
When you select the file you see the class definition for the activity you created.

simpledraw/AndroidManifest.xml

Steps for Creating Emulator And Run :

To run your app on the emulator you need to first create an Android Virtual Device (AVD). An AVD is a device configuration for the Android emulator that allows you to model a specific device.

Graphical Layout



Code for Login

```
package com.example.app3calculator;

import android.app.Activity;

import android.os.Bundle;

import android.text.Editable;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

public class MainActivity extends Activity implements View.OnClickListener {

Button one, two, three, four, five, six, seven, eight, nine, zero, add,

sub, mul, div, cancel, equal;

EditText disp;
```

```
int op1;
int op2;
String optr;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
one = (Button) findViewById(R.id.Button08);
two = (Button) findViewById(R.id.Button15);
three = (Button) findViewById(R.id.Button14);
four = (Button) findViewById(R.id.Button03);
five = (Button) findViewById(R.id.Button02);
six = (Button) findViewById(R.id.Button01);
seven = (Button) findViewById(R.id.button1);
eight = (Button) findViewById(R.id.Button06);
nine = (Button) findViewById(R.id.Button05);
zero = (Button) findViewById(R.id.Button11);
add = (Button) findViewById(R.id.Button09);
sub = (Button) findViewById(R.id.Button13);
mul = (Button) findViewById(R.id.Button07);
div = (Button) findViewById(R.id.Button04);
cancel = (Button) findViewById(R.id.Button12);
equal = (Button) findViewById(R.id.Button10);
disp = (EditText) findViewById(R.id.editText1);
try {
one.setOnClickListener(this);
two.setOnClickListener(this);
```

```
three.setOnClickListener(this);
four.setOnClickListener(this);
five.setOnClickListener(this);
six.setOnClickListener(this);
seven.setOnClickListener(this);
eight.setOnClickListener(this);
nine.setOnClickListener(this);
zero.setOnClickListener(this);
cancel.setOnClickListener(this);
add.setOnClickListener(this);
sub.setOnClickListener(this);
mul.setOnClickListener(this);
div.setOnClickListener(this);
equal.setOnClickListener(this);
} catch (Exception e) {}
}
public void operation() {
if (optr.equals("+")) {
op2 = Integer.parseInt(disp.getText().toString());
disp.setText("");
op1 = op1 + op2;
disp.setText(Integer.toString(op1));
} else if (optr.equals("-")) {
op2 = Integer.parseInt(disp.getText().toString());
disp.setText("");
op1 = op1 - op2;
disp.setText(Integer.toString(op1));
```

```
} else if (optr.equals("*")) {
op2 = Integer.parseInt(disp.getText().toString());
disp.setText("");
op1 = op1 * op2;
disp.setText(Integer.toString(op1));
} else if (optr.equals("/")) {
op2 = Integer.parseInt(disp.getText().toString());
disp.setText("");
op1 = op1 / op2;
disp.setText(Integer.toString(op1));
}
}
@Override
public void onClick(View arg0) {
Editable str = disp.getText();
switch (arg0.getId()) {
case R.id.Button08:
if (op2 != 0) {
op2 = 0;
disp.setText("");
}
str = str.append(one.getText());
disp.setText(str);
break;
case R.id.Button15:
if (op2 != 0) {
op2 = 0;
```

```
disp.setText("");
}
str = str.append(two.getText());
disp.setText(str);
break;
case R.id.Button14:
if (op2 != 0) {
op2 = 0;
disp.setText("");
}
str = str.append(three.getText());
disp.setText(str);
break;
case R.id.Button03:
if (op2 != 0) {
op2 = 0;
disp.setText("");
}
str = str.append(four.getText());
disp.setText(str);
break;
case R.id.Button02:
if (op2 != 0) {
op2 = 0;
disp.setText("");
}
str = str.append(five.getText());
```

```
disp.setText(str);
break;
case R.id.Button01:
if (op2 != 0) {
op2 = 0;
disp.setText("");
}
str = str.append(six.getText());
disp.setText(str);
break;
case R.id.button1:
if (op2 != 0) {
op2 = 0;
disp.setText("");
}
str = str.append(seven.getText());
disp.setText(str);
break;
case R.id.Button06:
if (op2 != 0) {
op2 = 0;
disp.setText("");
}
str = str.append(eight.getText());
disp.setText(str);
break;
case R.id.Button05:
```

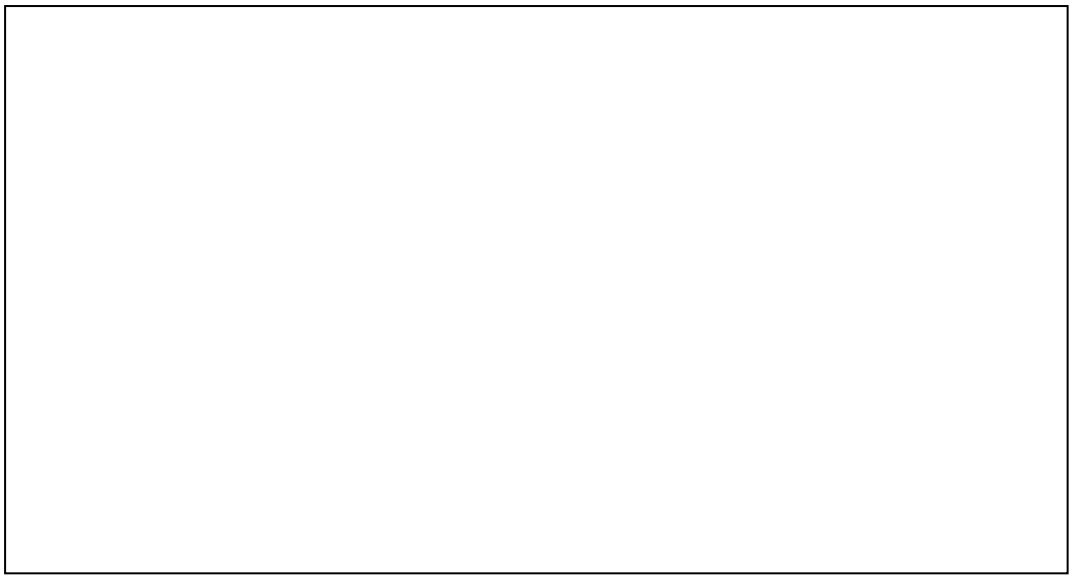
```
if (op2 != 0) {
op2 = 0;
disp.setText("");
}
str = str.append(nine.getText());
disp.setText(str);
break;
case R.id.Button11:
if (op2 != 0) {
op2 = 0;
disp.setText("");
}
str = str.append(zero.getText());
disp.setText(str);
break;
case R.id.editText1:
op1 = 0;
op2 = 0;
disp.setText("");
break;
case R.id.Button09:
optr = "+";
if (op1 == 0) {
op1 = Integer.parseInt(disp.getText().toString());
disp.setText("");
} else if (op2 != 0) {
op2 = 0;
```

```
disp.setText("");
} else {
op2 = Integer.parseInt(disp.getText().toString());
disp.setText("");
op1 = op1 + op2;
disp.setText("Result : " + Integer.toString(op1));
}
break;
case R.id.Button13:
optr = "-";
if (op1 == 0) {
op1 = Integer.parseInt(disp.getText().toString());
disp.setText("");
} else if (op2 != 0) {
op2 = 0;
disp.setText("");
} else {
op2 = Integer.parseInt(disp.getText().toString());
disp.setText("");
op1 = op1 - op2;
disp.setText("Result : " + Integer.toString(op1));
}
break;
case R.id.Button07:
optr = "*";
if (op1 == 0) {
op1 = Integer.parseInt(disp.getText().toString());
```

```
disp.setText("");
} else if (op2 != 0) {
op2 = 0;
disp.setText("");
} else {
op2 = Integer.parseInt(disp.getText().toString());
disp.setText("");
op1 = op1 * op2;
disp.setText("Result : " + Integer.toString(op1));
}
break;
case R.id.Button04:
optr = "/";
if (op1 == 0) {
op1 = Integer.parseInt(disp.getText().toString());
disp.setText("");
} else if (op2 != 0) {
op2 = 0;
disp.setText("");
} else {
op2 = Integer.parseInt(disp.getText().toString());
disp.setText("");
op1 = op1 / op2;
disp.setText("Result : " + Integer.toString(op1));
}
break;
case R.id.Button10:
```

```
if (!optr.equals(null)) {
if (op2 != 0) {
if (optr.equals("+")) {
disp.setText(""); /* op1 = op1 + op2; */
disp.setText(Integer.toString(op1));
} else if (optr.equals("-")) {
disp.setText(""); /* op1 = op1 - op2; */
disp.setText(Integer.toString(op1));
} else if (optr.equals("*")) {
disp.setText(""); /* op1 = op1 * op2; */
disp.setText(Integer.toString(op1));
} else if (optr.equals("/")) {
disp.setText(""); /* op1 = op1 / op2; */
disp.setText(Integer.toString(op1));
}
} else {
operation();
}
}
break;
}
}
}
```

OUTPUT



RESULT:

Exp: No: 7	Implement an application that creates an alert upon receiving a message

Aim:

To develop a Android Application that creates an alert upon receiving a message.

Procedure:

1. Start the process.

2. Start the Eclipse IDE for developing mobile application.

3. Create a new project by : File ->New -> Project -> Android (Wizard) - > Android Application Project .

4. In New Android Application dialog to enter the required values for creating the new android application such as Application Name, Package name and Project Name.

5. Enter the Application name and then click Next.

6. In Under create activity wizard Add an activity to <*template*>, select Blank Activity and click Next.

7. Change the Activity Name to *Mainactivity* then Click the Finish button to create the project.

8. Android Project is created which contains the following file

simpledraw/res/layout/activity_mainactivity.xml

This is the XML layout file for the activity you added when you created the project with Android Studio.

simpledraw/src/com.exmaple.simpledraw/ Mainactivity.java

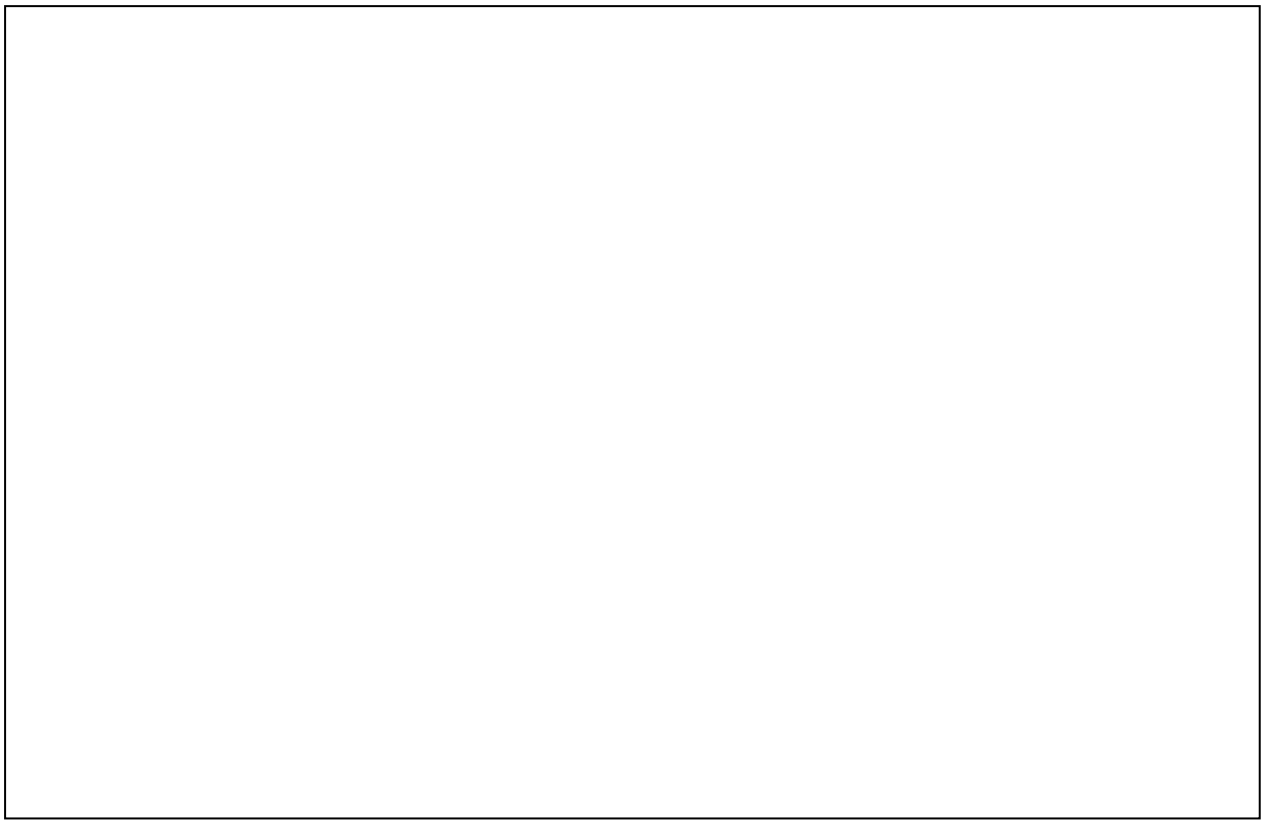
When you select the file you see the class definition for the activity you created.

simpledraw/AndroidManifest.xml

Steps for Creating Emulator And Run :

To run your app on the emulator you need to first create an Android Virtual Device (AVD). An AVD is a device configuration for the Android emulator that allows you to model a specific device.

Graphical Layout



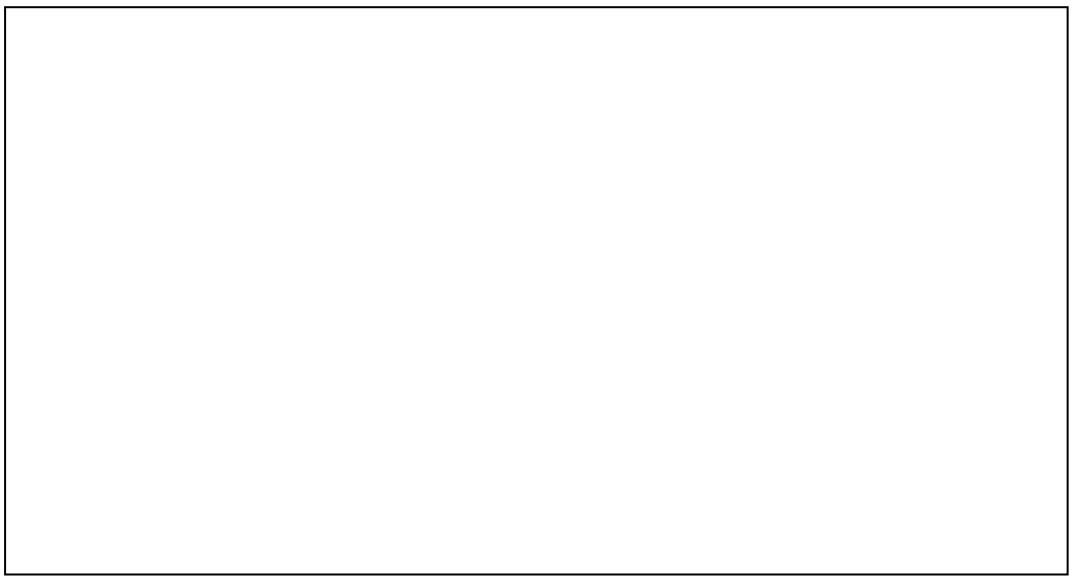
Code for Login

```
package com.example.exno10;


import android.app.Notification;

import android.app.NotificationManager;

import android.app.PendingIntent;

import android.content.Intent;

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;
```

```
public class MainActivity extends AppCompatActivity
{
    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.mipmap.ic_launcher).setCo
ntentIntent(pending).build();
                NotificationManager manager = (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);
                noti.flags |= Notification.FLAG_AUTO_CANCEL;
```

```
            manager.notify(0, noti);
        }
    });
  }
}
```

OUTPUT



RESULT:

Exp: No: 8	Develop a mobile application to send an email

Aim:

To develop a mobile application to send an email.

Procedure:

1. Start the process.

2. Start the Eclipse IDE for developing mobile application.

3. Create a new project by : File ->New -> Project -> Android (Wizard) - > Android Application Project .

4. In New Android Application dialog to enter the required values for creating the new android application such as Application Name, Package name and Project Name.

5. Enter the Application name and then click Next.

6. In Under create activity wizard Add an activity to <*template*>, select Blank Activity and click Next.

7. Change the Activity Name to *Mainactivity* then Click the Finish button to create the project.

8. Android Project is created which contains the following file

simpledraw/res/layout/activity_mainactivity.xml

This is the XML layout file for the activity you added when you created the project with Android Studio.

simpledraw/src/com.exmaple.simpledraw/ Mainactivity.java

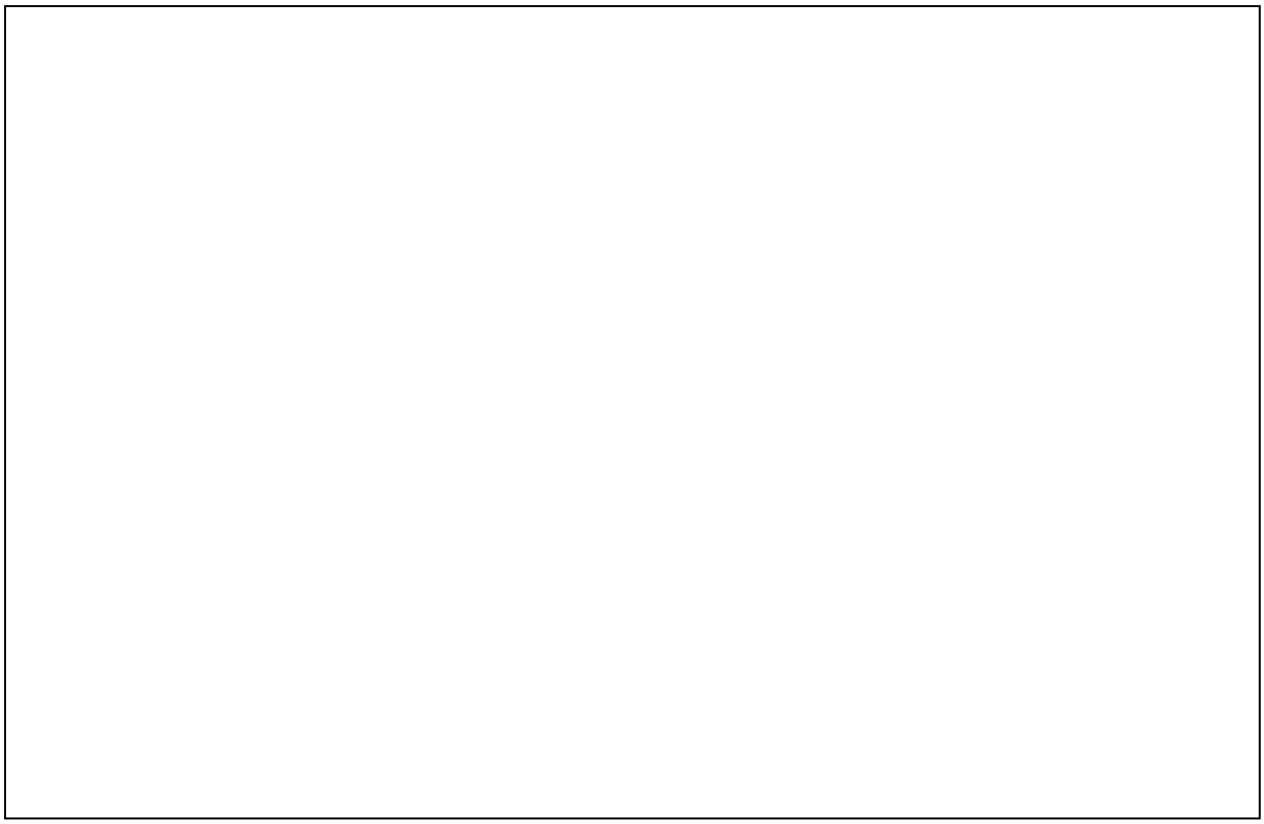
When you select the file you see the class definition for the activity you created.

simpledraw/AndroidManifest.xml

Steps for Creating Emulator And Run :

To run your app on the emulator you need to first create an Android Virtual Device (AVD). An AVD is a device configuration for the Android emulator that allows you to model a specific device.

Graphical Layout



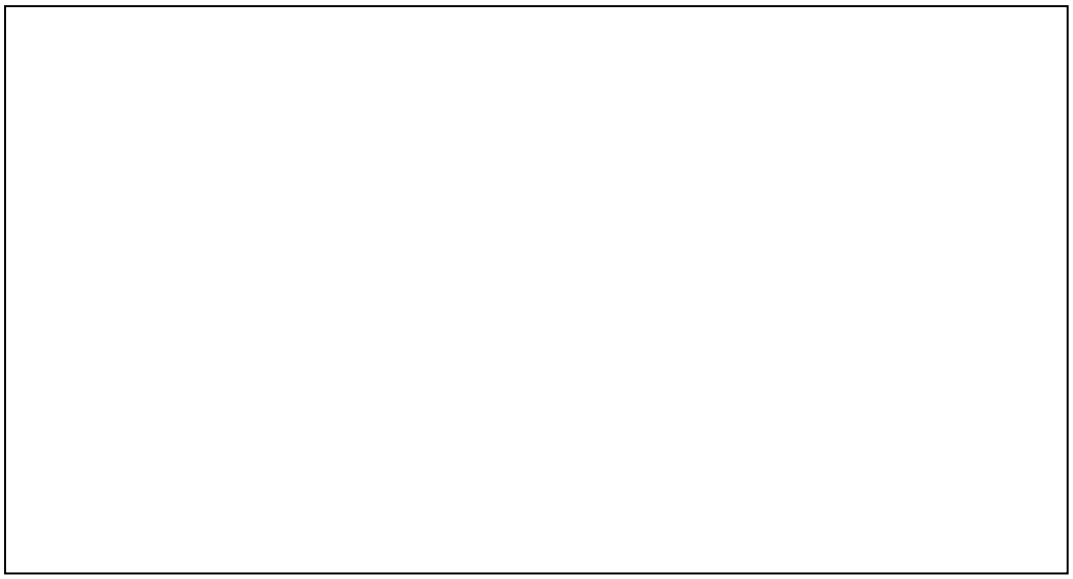
Code for Login

```
package com.example.tutorialspoint;

import android.net.Uri;

import android.os.Bundle;

import android.app.Activity;

import android.content.Intent;

import android.util.Log;

import android.view.Menu;

import android.view.View;

import android.widget.Button;

import android.widget.Toast;

public class MainActivity extends Activity {
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
Button startBtn = (Button) findViewById(R.id.sendEmail);
startBtn.setOnClickListener(new View.OnClickListener() {
public void onClick(View view) {
sendEmail();
}
});
}
protected void sendEmail() {
Log.i("Send email", "");
String[] TO = {""};
String[] CC = {""};
Intent emailIntent = new Intent(Intent.ACTION_SEND);
emailIntent.setData(Uri.parse("mailto:"));
emailIntent.setType("text/plain");
emailIntent.putExtra(Intent.EXTRA_EMAIL, TO);
emailIntent.putExtra(Intent.EXTRA_CC, CC);
emailIntent.putExtra(Intent.EXTRA_SUBJECT, "Your subject");
emailIntent.putExtra(Intent.EXTRA_TEXT, "Email message goes here");
try {
startActivity(Intent.createChooser(emailIntent, "Send mail..."));
finish();
Log.i("Finished sending email...", "");
} catch (android.content.ActivityNotFoundException ex) {
```

```
Toast.makeText(MainActivity.this, "There is no email client installed.",
Toast.LENGTH_SHORT).show();
}
}
}
```

OUTPUT



RESULT:

Exp: No: 9	Develop an application for Web view to display website

Aim:

To develop an mobile application for web view to display website.

Procedure:

1. Start the process.

2. Start the Eclipse IDE for developing mobile application.

3. Create a new project by : File ->New -> Project -> Android (Wizard) - > Android Application Project .

4. In New Android Application dialog to enter the required values for creating the new android application such as Application Name, Package name and Project Name.

5. Enter the Application name and then click Next.

6. In Under create activity wizard Add an activity to <*template*>, select Blank Activity and click Next.

7. Change the Activity Name to *Mainactivity* then Click the Finish button to create the project.

8. Android Project is created which contains the following file

simpledraw/res/layout/activity_mainactivity.xml

This is the XML layout file for the activity you added when you created the project with Android Studio.

simpledraw/src/com.exmaple.simpledraw/ Mainactivity.java

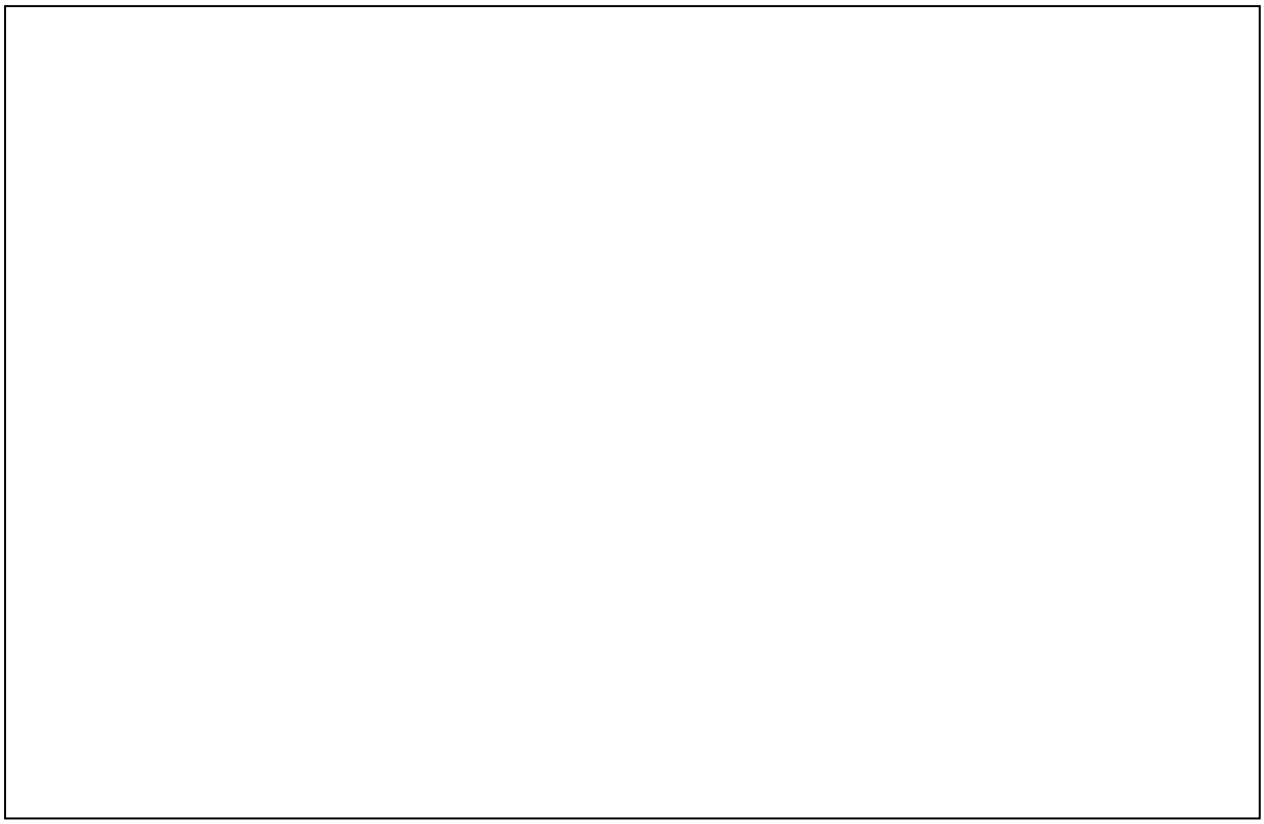
When you select the file you see the class definition for the activity you created.

simpledraw/AndroidManifest.xml

Steps for Creating Emulator And Run :

To run your app on the emulator you need to first create an Android Virtual Device (AVD). An AVD is a device configuration for the Android emulator that allows you to model a specific device.

Graphical Layout

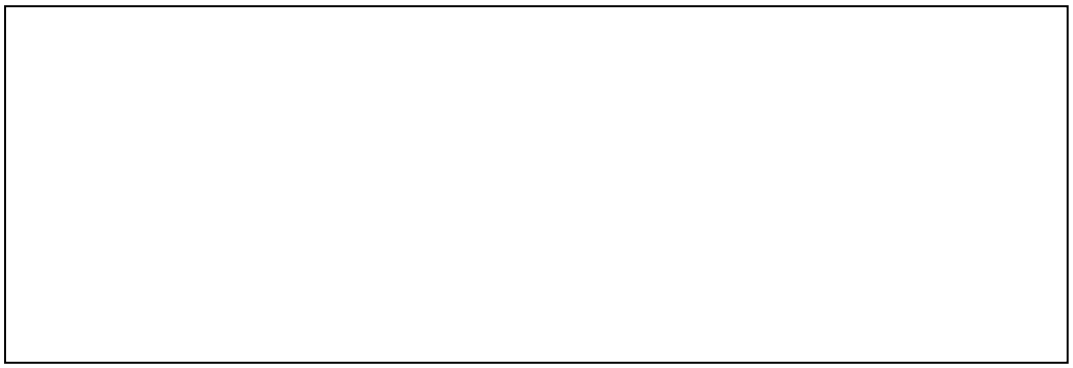


Code for Login

```
package com.example.exno9;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.webkit.WebResourceRequest;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.Button;
import android.widget.EditText;
public class MainActivity extends AppCompatActivity {
EditText url;
```

```
Button search;
WebView site;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
url=(EditText)findViewById(R.id.editText);
search=(Button)findViewById(R.id.button);
search.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
String web = url.getText().toString();
site = (WebView) findViewById(R.id.webView);
site.setWebViewClient(new WebViewClient());
site.loadUrl(web);
 }
 });
 }
}
```

OUTPUT



RESULT:

Exp: No: 10	Develop an application to display a location on Map using Map view

Aim:

To develop an mobile application to display a location on Map using Map view.

Procedure:

1. Start the process.

2. Start the Eclipse IDE for developing mobile application.

3. Create a new project by : File ->New -> Project -> Android (Wizard) - > Android Application Project .

4. In New Android Application dialog to enter the required values for creating the new android application such as Application Name, Package name and Project Name.

5. Enter the Application name and then click Next.

6. In Under create activity wizard Add an activity to <*template*>, select Blank Activity and click Next.

7. Change the Activity Name to *Mainactivity* then Click the Finish button to create the project.

8. Android Project is created which contains the following file

simpledraw/res/layout/activity_mainactivity.xml

This is the XML layout file for the activity you added when you created the project with Android Studio.

simpledraw/src/com.exmaple.simpledraw/ Mainactivity.java

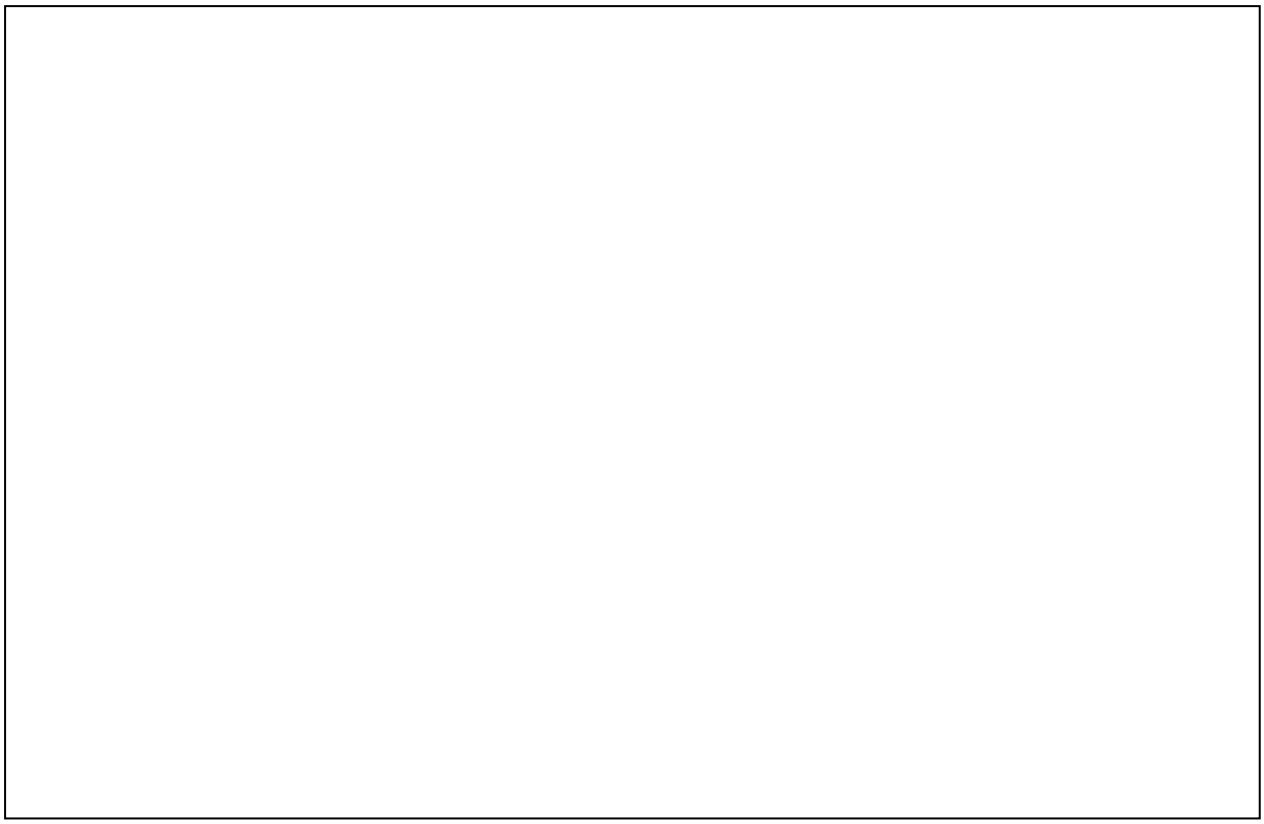
When you select the file you see the class definition for the activity you created.

simpledraw/AndroidManifest.xml

Steps for Creating Emulator And Run :

To run your app on the emulator you need to first create an Android Virtual Device (AVD). An AVD is a device configuration for the Android emulator that allows you to model a specific device.

Graphical Layout



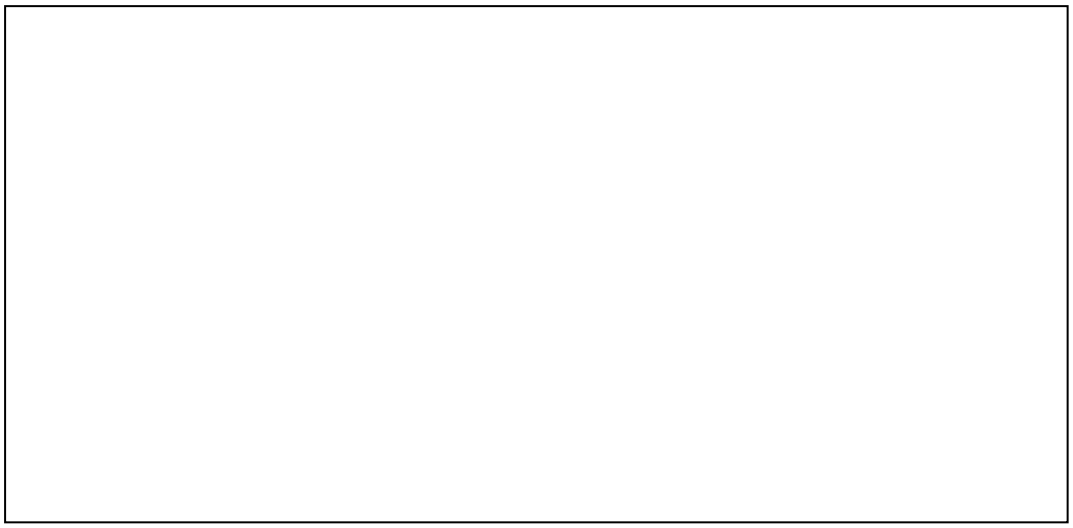
Code for Login

```
package com.example.exno10;

import android.location.Address;

import android.location.Geocoder;

import android.support.v4.app.FragmentActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.EditText;

import com.google.android.gms.maps.CameraUpdateFactory;

import com.google.android.gms.maps.GoogleMap;

import com.google.android.gms.maps.MapView;

import com.google.android.gms.maps.OnMapReadyCallback;
```

```
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
import java.io.IOException;
import java.util.List;
public class MapsActivity extends FragmentActivity implements
OnMapReadyCallback {
 private GoogleMap mMap;
 MapView view;
 @Override
 protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_maps);
 view= (MapView) findViewById(R.id.mapview);
 view.onCreate(savedInstanceState);
 view.getMapAsync(this);
 }
 @Override
 protected void onResume() {
 view.onResume();
 super.onResume();
 }
 @Override
 public void onMapReady(GoogleMap googleMap) {
 mMap = googleMap;
 mMap.addMarker(new MarkerOptions().position(new LatLng(0,
0)).title("Marker"));
 mMap.setMyLocationEnabled(true);
```

```
 }
public void onMapSearch(View view) {
 EditText locationSearch = (EditText) findViewById (R.id.editText);
 String location = locationSearch.getText().toString();
 List<Address> addressList = null;
 if (location != null || !location.equals("")) {
 Geocoder geocoder = new Geocoder(this);
 try {
 addressList = geocoder.getFromLocationName(location, 1);
 } catch (IOException e) {
 e.printStackTrace();
 }
 Address address = addressList.get(0);
 LatLng latLng = new LatLng(address.getLatitude(),
address.getLongitude());
 mMap.addMarker(new
MarkerOptions().position(latLng).title(location));
 mMap.animateCamera(CameraUpdateFactory.newLatLng(latLng));
 }
 }
}
```

OUTPUT



RESULT: