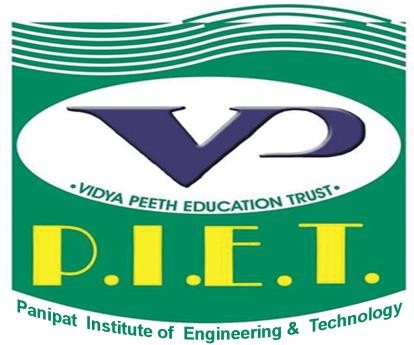
**Panipat Institute of Engineering & Technology, Samalkha, Panipat**



**Computer Science & Engineering Department**

Practical File of Mobile Apps Development Sub Code: CSE- 406N

Submitted To: Submitted By:

## Mr. Shiraz Khurana Karan Abrol

Assistant Professor 2817060

(CSE Department)

**Affiliated to**

**Kurukshetra University Kurukshetra, India**

**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Title of the Practical** | **Date** | **Page No.** | **Remarks** |
| 1. | Develop an application that uses GUI components, Font and colours. |  |  |  |
| 2. | Develop an application that uses Layout Managers and event listeners. |  |  |  |
| 3. | Develop a native calculator application. |  |  |  |
| 4. | Write an application that draws basic graphical primitives on the screen. |  |  |  |
| 5. | Develop a real-life application that makes use of databases. |  |  |  |
| 6. | Implement an application that implements multithreading. |  |  |  |
| 7. | Develop a native application that uses GPS location information. |  |  |  |
| 8. | Implement an application that creates an alert upon receiving a message. |  |  |  |
| 9. | Write a mobile application that creates alarm clock. |  |  |  |
| 10. | Implement an application that writes data to the SD card. |  |  |  |

# Practical No.-1

### Aim: Develop an application that uses GUI components, Fonts and colors.

**Hardware Requirements & Software Requirements:**

* Android Studio3.0
* Windows 7/8/10 OS
* RAM 4 GB
* Hard disk 10 GB,
* Emulator/mobile phones etc;

### Procedure:

* 1. Open android studio and select new android project.
  2. Give project name and select next.
  3. Choose the minimum target API version and select next.
  4. Enter the package name, and its domain. Package name must be of two words separated by comma and then click on finish.
  5. Go to package explorer in the left-hand side and select your project.
  6. Go to res folder and select layout. Double click the main.xml file.
  7. Now you can see the Graphics layout window.

**Source code:**

### Front-end:

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

<LinearLayout [xmlns:android="http://schemas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android) android:layout\_width="fill\_parent" android:layout\_height="fill\_parent" android:orientation="vertical" >

<TextView android:id="@+id/textView1" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:layout\_margin="20sp"

android:gravity="center" android:text="HELLO WORLD" android:textSize="20sp" android:textStyle="bold" />

<Button

android:id="@+id/button1" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:gravity="center" android:text="Change font size" android:textSize="20sp" />

<Button android:id="@+id/button2"

android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:gravity="center" android:text="Change color" android:textSize="20sp" />

<Button android:id="@+id/button3"

android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:gravity="center" android:text="Change font" android:textSize="20sp" />

</LinearLayout>

### Back-end:

import android.app.Activity; import android.graphics.Color; import android.graphics.Typeface; import android.os.Bundle;

import android.view.View; import android.widget.Button; import android.widget.TextView;

public class AndroidActivity extends Activity { float font =24;

int i=1; @Override

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

final TextView t1=(TextView) findViewById(R.id.textView1); Button b1 = (Button) findViewById(R.id.button1); b1.setOnClickListener(new View.OnClickListener() {

public void onClick(View view) {

t1.setTextSize(font); font=font+4; if(font==40)

font=20;

}

});

Button b2 = (Button) findViewById(R.id.button2); b2.setOnClickListener(new View.OnClickListener() { public void onClick(View view) {

switch(i)

{

case 1: t1.setTextColor(Color.parseColor("#0000FF")); break;

case 2: t1.setTextColor(Color.parseColor("#00FF00")); break;

case 3: t1.setTextColor(Color.parseColor("#FF0000")); break;

case 4: t1.setTextColor(Color.parseColor("#800000")); break;

} i++;

if(i==5) i=1;

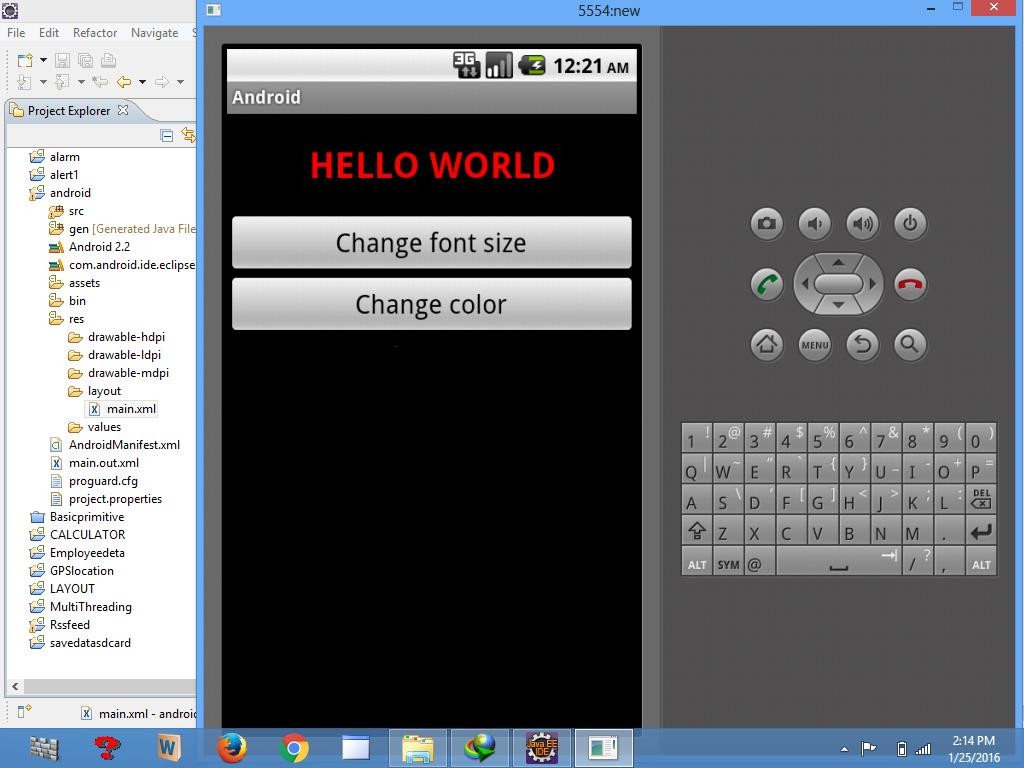
}

});

}

}

**Output:**



# Practical No.-2

### Aim: Develop an application that uses layout managers and event listeners.

**Hardware Requirements & Software Requirements:**

* Android Studio3.0
* Windows 7/8/10 OS
* RAM 4 GB
* Hard disk 10 GB,
* Emulator/mobile phones etc;

### Procedure:

1. Open android studio and select new android project.
2. Give project name and select next.
3. Choose the minimum target API version and select next.
4. Enter the package name, and its domain. Package name must be of two words separated by comma and then click on finish.
5. Go to package explorer in the left-hand side and select your project.
6. Go to res folder and select layout. Double click the main.xml file.
7. Now you can see the Graphics layout window.

**Source code:**

**Front-end:**

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

<RelativeLayout [xmlns:android="http://schemas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android) android:id="@+id/relativeLayout1" android:layout\_width="fill\_parent" android:layout\_height="fill\_parent" >

<LinearLayout android:id="@+id/linearLayout1" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignParentLeft="true"

android:layout\_alignParentRight="true" android:layout\_alignParentTop="true" >

<TextView android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_gravity="center" android:text="ADDITION" android:textSize="20dp" >

</TextView>

</LinearLayout>

<LinearLayout android:id="@+id/linearLayout2" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignParentLeft="true" android:layout\_alignParentRight="true" android:layout\_below="@+id/linearLayout1" >

<TextView android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="ENTER NO 1" >

</TextView>

<EditText android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_weight="0.20" android:id="@+id/edittext1" android:inputType="number">

</EditText>

</LinearLayout>

<LinearLayout android:id="@+id/linearLayout3" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignParentLeft="true" android:layout\_alignParentRight="true" android:layout\_below="@+id/linearLayout2" >

<TextView android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="ENTER NO 2" >

</TextView>

<EditText android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_weight="0.20" android:id="@+id/edittext2"

android:inputType="number">

</EditText>

</LinearLayout>

<LinearLayout android:id="@+id/linearLayout4" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignParentLeft="true" android:layout\_alignParentRight="true" android:layout\_below="@+id/linearLayout3" >

<Button android:layout\_width="wrap\_content" android:id="@+id/button1" android:layout\_height="wrap\_content" android:text="Addition" android:layout\_weight="0.50" />

<Button android:layout\_width="wrap\_content" android:id="@+id/button3" android:layout\_height="wrap\_content" android:text="subtraction" android:layout\_weight="0.50" />

<Button android:layout\_width="wrap\_content" android:id="@+id/button2" android:layout\_height="wrap\_content" android:text="CLEAR" android:layout\_weight="0.50" />

</LinearLayout>

<View android:layout\_height="2px"

android:layout\_width="fill\_parent" android:layout\_below="@+id/linearLayout4" android:background="#DDFFDD"/>

</RelativeLayout>

### Back end:

package layout.ne;

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

import android.view.View.OnClickListener; import android.widget.Button;

import android.widget.EditText; import android.widget.Toast;

public class LAYOUTActivity extends Activity {

/\*\* Called when the activity is first created. \*/

EditText txtData1,txtData2; float num1,num2,result1,result2; @Override

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

Button add = (Button) findViewById(R.id.button1); add.setOnClickListener(new OnClickListener() { public void onClick(View v) {

try

{

txtData1 = (EditText) findViewById(R.id.edittext1); txtData2 = (EditText) findViewById(R.id.edittext2); num1 = Float.parseFloat(txtData1.getText().toString()); num2 = Float.parseFloat(txtData2.getText().toString()); result1=num1+num2;

Toast.makeText(getBaseContext(),"ANSWER:"+result1,Toast.LENGTH\_S HORT).show();

}

catch(Exception e)

{

Toast.makeText(getBaseContext(), e.getMessage(), Toast.LENGTH\_SHORT).show();

}

}

});

Button sub = (Button) findViewById(R.id.button3); sub.setOnClickListener(new OnClickListener() { public void onClick(View v) {

try

{

txtData1 = (EditText) findViewById(R.id.edittext1); txtData2 = (EditText) findViewById(R.id.edittext2); num1 = Float.parseFloat(txtData1.getText().toString()); num2 = Float.parseFloat(txtData2.getText().toString()); result2=num1-num2;

Toast.makeText(getBaseContext(),"ANSWER:"+result2,Toast.LENGTH\_S HORT).show();

}

catch(Exception e)

{

Toast.makeText(getBaseContext(), e.getMessage(), Toast.LENGTH\_SHORT).show();

}

}

});

Button clear = (Button) findViewById(R.id.button2);

clear.setOnClickListener(new OnClickListener() { public void onClick(View v) {

try

{

txtData1.setText(""); txtData2.setText("");

}

catch(Exception e)

{

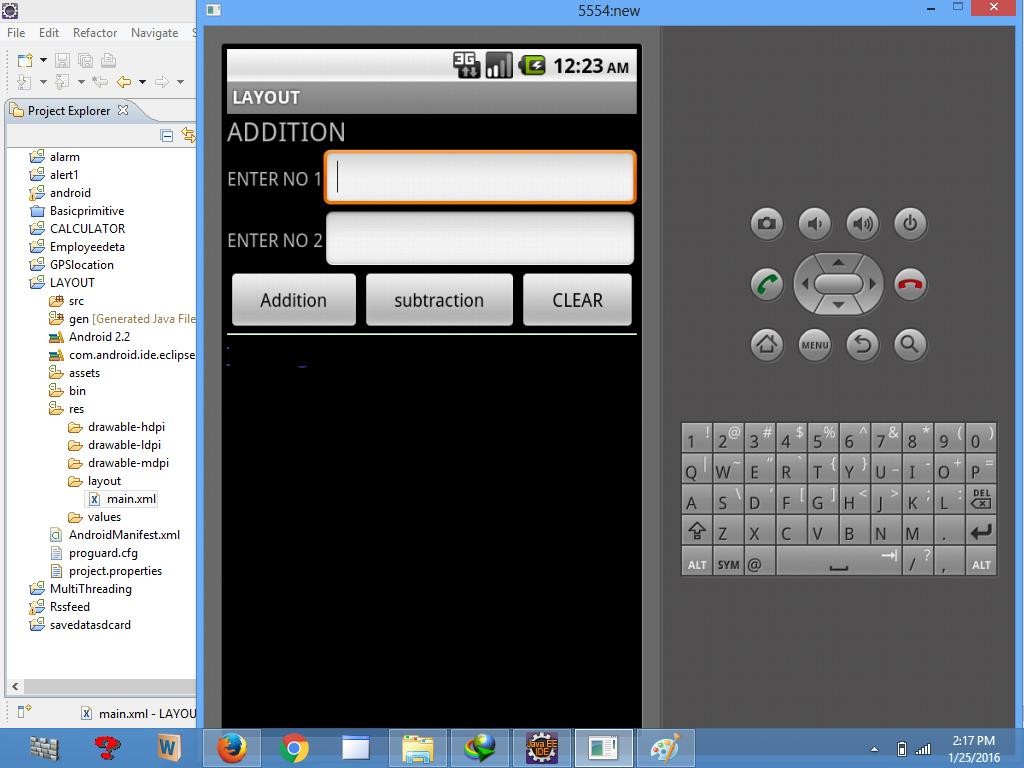
Toast.makeText(getBaseContext(), e.getMessage(), Toast.LENGTH\_SHORT).show();

}

} });

}

}

**Output:**

# Practical No.-3

### Aim: Develop a native calculator application.

**Hardware Requirements & Software Requirements:**

* Android Studio3.0
* Windows 7/8/10 OS
* RAM 4 GB
* Hard disk 10 GB,
* Emulator/mobile phones etc;

### Procedure:

1. Open android studio and select new android project.
2. Give project name and select next.
3. Choose the minimum target API version and select next.
4. Enter the package name, and its domain. Package name must be of two words separated by comma and then click on finish.
5. Go to package explorer in the left-hand side and select your project.
6. Go to res folder and select layout. Double click the main.xml file.
7. Now you can see the Graphics layout window.

**Source code:**

### Front-end:

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

<LinearLayout xmlns:android[="http://schemas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android) android:orientation="vertical" android:layout\_width="fill\_parent" android:layout\_height="fill\_parent">

<LinearLayout android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:id="@+id/linearLayout1" android:layout\_marginLeft="10pt"

android:layout\_marginRight="10pt" android:layout\_marginTop="3pt">

<EditText android:layout\_weight="1"

android:layout\_height="wrap\_content" android:layout\_marginRight="5pt" android:id="@+id/etNum1" android:layout\_width="match\_parent" android:inputType="numberDecimal">

</EditText>

<EditText android:layout\_height="wrap\_content" android:layout\_weight="1" android:layout\_marginLeft="5pt" android:id="@+id/etNum2" android:layout\_width="match\_parent" android:inputType="numberDecimal">

</EditText>

</LinearLayout>

<LinearLayout android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:id="@+id/linearLayout2" android:layout\_marginTop="3pt" android:layout\_marginLeft="5pt" android:layout\_marginRight="5pt">

<Button android:layout\_height="wrap\_content" android:layout\_width="match\_parent" android:layout\_weight="1" android:text="+" android:textSize="15pt" android:id="@+id/btnAdd">

</Button>

<Button android:layout\_height="wrap\_content" android:layout\_width="match\_parent" android:layout\_weight="1" android:text="-" android:textSize="15pt" android:id="@+id/btnSub">

</Button>

<Button android:layout\_height="wrap\_content" android:layout\_width="match\_parent" android:layout\_weight="1" android:text="\*"

android:textSize="15pt" android:id="@+id/btnMult">

</Button>

<Button android:layout\_height="wrap\_content" android:layout\_width="match\_parent" android:layout\_weight="1" android:text="/" android:textSize="15pt" android:id="@+id/btnDiv">

</Button>

</LinearLayout>

<TextView android:layout\_height="wrap\_content" android:layout\_width="match\_parent" android:layout\_marginLeft="5pt" android:layout\_marginRight="5pt" android:textSize="12pt" android:layout\_marginTop="3pt" android:id="@+id/tvResult" android:gravity="center\_horizontal">

</TextView>

</LinearLayout>

### Back end:

package CALCU.CALU;

import android.app.Activity; import android.os.Bundle; 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 CALCULATORActivity extends Activity implements OnClickListener {

EditText input1; EditText input2; Button addition; Button subtraction; Button multiplication; Button division; TextView tvResult; String oper = ""; @Override

public void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState);

setContentView(R.layout.main);

input1 = (EditText) findViewById(R.id.etNum1); input2 = (EditText) findViewById(R.id.etNum2); addition = (Button) findViewById(R.id.btnAdd); subtraction = (Button) findViewById(R.id.btnSub);

multiplication = (Button) findViewById(R.id.btnMult); division = (Button) findViewById(R.id.btnDiv); tvResult = (TextView) findViewById(R.id.tvResult);

// set a listener addition.setOnClickListener(this); subtraction.setOnClickListener(this); multiplication.setOnClickListener(this); division.setOnClickListener(this);

}

@Override

public void onClick(View v) {

// TODO Auto-generated method stub float num1 = 0;

float num2 = 0; float result = 0;

// check if the fields are empty

if (TextUtils.isEmpty(input1.getText().toString())

|| TextUtils.isEmpty(input2.getText().toString())) { return;

}

// read EditText and fill variables with numbers num1 = Float.parseFloat(input1.getText().toString()); num2 = Float.parseFloat(input2.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.btnAdd: oper = "+";

result = num1 + num2; break;

case R.id.btnSub: oper = "-";

result = num1 - num2; break;

case R.id.btnMult: oper = "\*";

result = num1 \* num2; break;

case R.id.btnDiv: oper = "/";

result = num1 / num2;

break; default:

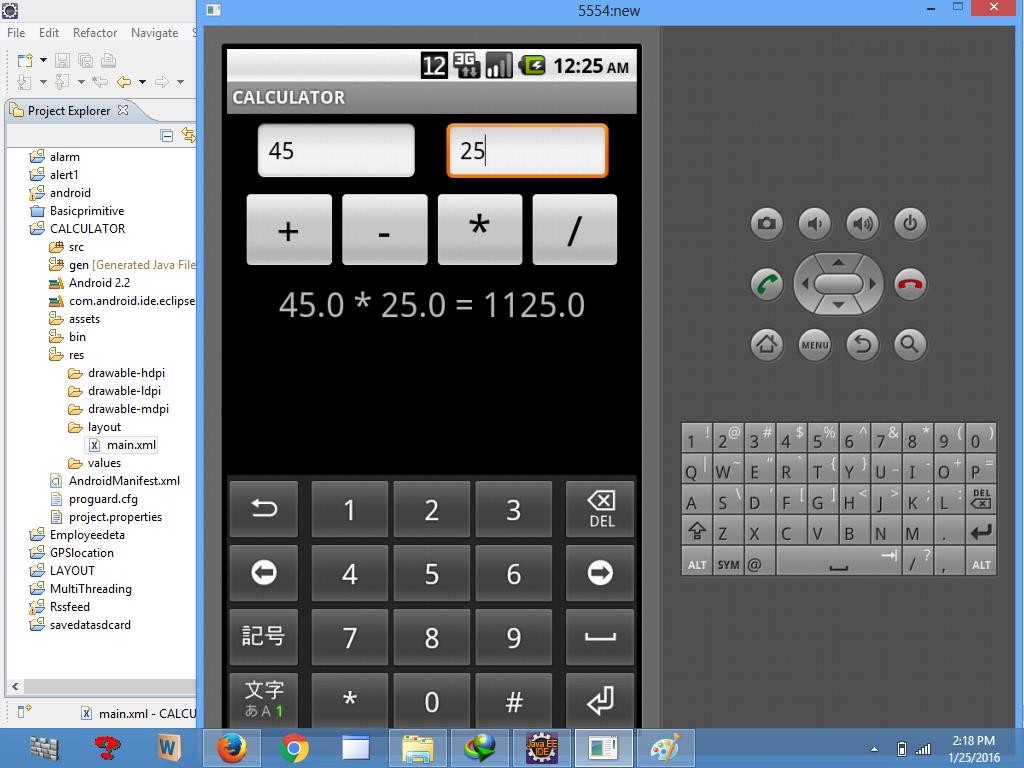
break;

}

// form the output line

tvResult.setText(num1 + " " + oper + " " + num2 + " = " + result);} }

**Output:**



# Practical No.-4

### Aim: Write an application that draws graphical primitives on the screen.

**Hardware Requirements & Software Requirements:**

* Android Studio3.0
* Windows 7/8/10 OS
* RAM 4 GB
* Hard disk 10 GB,
* Emulator/mobile phones etc;

### Procedure:

1. Open android studio and select new android project.
2. Give project name and select next.
3. Choose the minimum target API version and select next.
4. Enter the package name, and its domain. Package name must be of two words separated by comma and then click on finish.
5. Go to package explorer in the left-hand side and select your project.
6. Go to res folder and select layout. Double click the main.xml file.
7. Now you can see the Graphics layout window.

**Source code:**

### Front-end:

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

<LinearLayout [xmlns:android="http://schemas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android) android:orientation="vertical" android:layout\_width="fill\_parent" android:layout\_height="fill\_parent">

<LinearLayout android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:id="@+id/linearLayout1" android:layout\_marginLeft="10pt"

android:layout\_marginRight="10pt" android:layout\_marginTop="3pt">

<EditText android:layout\_weight="1"

android:layout\_height="wrap\_content" android:layout\_marginRight="5pt" android:id="@+id/etNum1" android:layout\_width="match\_parent" android:inputType="numberDecimal">

</EditText>

<EditText android:layout\_height="wrap\_content" android:layout\_weight="1" android:layout\_marginLeft="5pt" android:id="@+id/etNum2" android:layout\_width="match\_parent" android:inputType="numberDecimal">

</EditText>

</LinearLayout>

### Back end:

package Basic.primitive; import android.app.Activity;

import android.content.Context; import android.graphics.Canvas; import android.graphics.Color; import android.graphics.Paint; import android.os.Bundle; import android.view.View;

public class BasicprimitiveActivity extends Activity {

/\*\* Called when the activity is first created. \*/ @Override

public void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(new myview(this));

}

private class myview extends View

{public myview(Context context){ super(context);

}

@Override

protected void onDraw(Canvas canvas)

{

super.onDraw(canvas); Paint paint=new Paint(); paint.setTextSize(40);

paint.setColor(Color.GREEN);

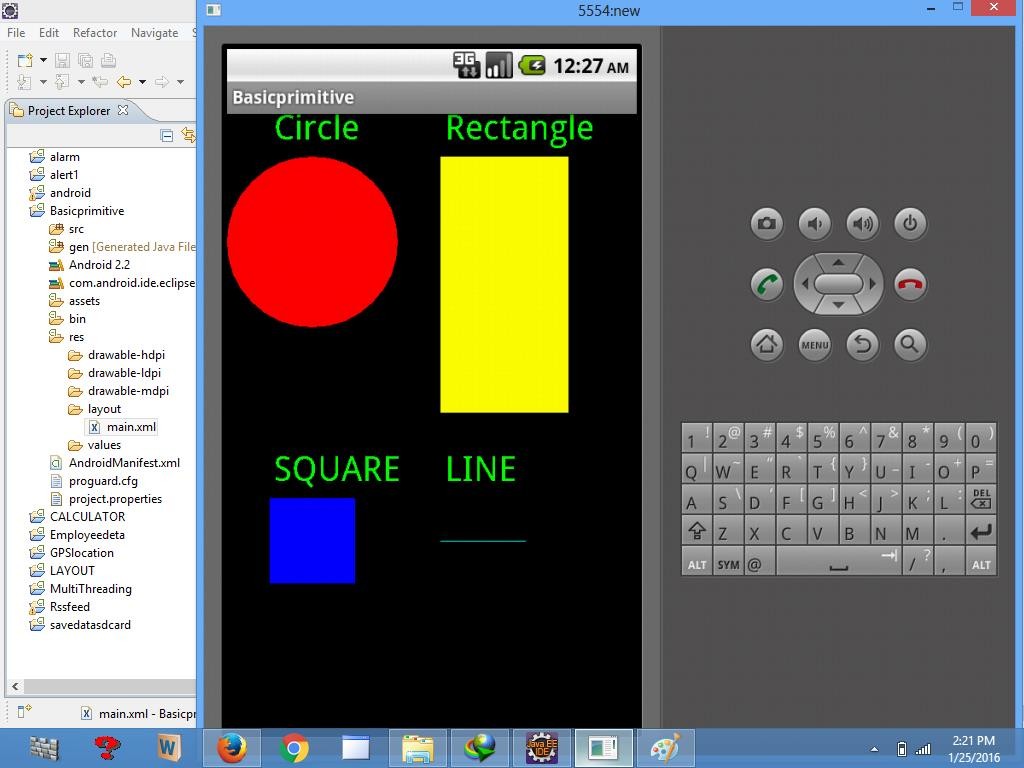
canvas.drawText("Circle", 55, 30, paint); paint.setColor(Color.RED); canvas.drawCircle(100, 150,100, paint); paint.setColor(Color.GREEN); canvas.drawText("Rectangle", 255, 30, paint); paint.setColor(Color.YELLOW); canvas.drawRect(250, 50,400,350, paint); paint.setColor(Color.GREEN); canvas.drawText("SQUARE", 55, 430, paint); paint.setColor(Color.BLUE); canvas.drawRect(50, 450,150,550, paint); paint.setColor(Color.GREEN); canvas.drawText("LINE", 255, 430, paint); paint.setColor(Color.CYAN); canvas.drawLine(250, 500, 350, 500, paint);

}

}

}

**Output:**



# Practical No.-5

### Aim: Develop a real-life application that makes use of database.

**Hardware Requirements & Software Requirements:**

* Android Studio3.0
* Windows 7/8/10 OS
* RAM 4 GB
* Hard disk 10 GB,
* Emulator/mobile phones etc;

### Procedure:

1. Open android studio and select new android project.
2. Give project name and select next.
3. Choose the minimum target API version and select next.
4. Enter the package name, and its domain. Package name must be of two words separated by comma and then click on finish.
5. Go to package explorer in the left-hand side and select your project.
6. Go to res folder and select layout. Double click the main.xml file.
7. Now you can see the Graphics layout window.

**Source code:**

### Front-end:

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

<AbsoluteLayout [xmlns:android="http://schemas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android) android:id="@+id/myLayout"

android:stretchColumns="0" android:layout\_width="fill\_parent" android:layout\_height="fill\_parent">

<TextView android:text="@string/title" android:layout\_x="110dp" android:layout\_y="10dp" android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"/>

<TextView android:text="@string/empid" android:layout\_x="30dp" android:layout\_y="50dp" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"/>

<EditText android:id="@+id/editEmpid" android:inputType="number" android:layout\_x="150dp" android:layout\_y="50dp" android:layout\_width="150dp" android:layout\_height="40dp"/>

<TextView android:text="@string/name" android:layout\_x="30dp" android:layout\_y="100dp" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"/>

<EditText android:id="@+id/editName" android:inputType="text" android:layout\_x="150dp" android:layout\_y="100dp" android:layout\_width="150dp" android:layout\_height="40dp"/>

<TextView android:text="@string/salary" android:layout\_x="30dp" android:layout\_y="150dp" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"/>

<EditText android:id="@+id/editsalary" android:inputType="number" android:layout\_x="150dp" android:layout\_y="150dp" android:layout\_width="150dp" android:layout\_height="40dp"/>

<Button android:id="@+id/btnAdd" android:text="@string/add" android:layout\_x="30dp" android:layout\_y="200dp" android:layout\_width="130dp" android:layout\_height="40dp"/>

<Button android:id="@+id/btnDelete" android:text="@string/delete" android:layout\_x="160dp" android:layout\_y="200dp" android:layout\_width="130dp" android:layout\_height="40dp"/>n

<Button android:id="@+id/btnModify"

android:text="@string/modify" android:layout\_x="30dp" android:layout\_y="250dp" android:layout\_width="130dp" android:layout\_height="40dp"/>

<Button android:id="@+id/btnView" android:text="@string/view" android:layout\_x="160dp" android:layout\_y="250dp" android:layout\_width="130dp" android:layout\_height="40dp"/>

<Button android:id="@+id/btnViewAll" android:text="@string/view\_all" android:layout\_x="85dp" android:layout\_y="300dp" android:layout\_width="150dp" android:layout\_height="40dp"/>

</AbsoluteLayout>

//\*Go to values folder and select string.xml file,Replace the code below\*//

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

<resources>

<string name="app\_name">Employee detail1</string>

<string name="hello">Hello World, Employee detail Activity!</string>

<string name="title">Employee Details</string>

<string name="empid">Enter Employee ID: </string>

<string name="name">Enter Name: </string>

<string name="salary">Enter salary: </string>

<string name="add">Add Employee</string>

<string name="delete">Delete Employee</string>

<string name="modify">Modify Employee</string>

<string name="view">View Employee</string>

<string name="view\_all">View All Employee</string>

</resources>

//\*Now select mainactivity.java file and type the following code.In my coding maniactivity name is EmployeedetailActivity\*//

### Back end:

package employee.detail;

//import android.R;

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 EmployeedetailActivity extends Activity implements OnClickListener {

EditText editEmpid,editName,editsalary;

Button btnAdd,btnDelete,btnModify,btnView,btnViewAll; SQLiteDatabase db;

/\*\* Called when the activity is first created. \*/ @Override

public void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState); setContentView(R.layout.main); editEmpid=(EditText)findViewById(R.id.editEmpid); editName=(EditText)findViewById(R.id.editName); editsalary=(EditText)findViewById(R.id.editsalary); btnAdd=(Button)findViewById(R.id.btnAdd); btnDelete=(Button)findViewById(R.id.btnDelete); btnModify=(Button)findViewById(R.id.btnModify); btnView=(Button)findViewById(R.id.btnView); btnViewAll=(Button)findViewById(R.id.btnViewAll); btnAdd.setOnClickListener(this); btnDelete.setOnClickListener(this); btnModify.setOnClickListener(this); btnView.setOnClickListener(this); btnViewAll.setOnClickListener(this); db=openOrCreateDatabase("EmployeeDB", Context.MODE\_PRIVATE, null);

db.execSQL("CREATE TABLE IF NOT EXISTS employee(empid VARCHAR,name

VARCHAR,salary VARCHAR);");

}

public void onClick(View view)

{

if(view==btnAdd)

{

if(editEmpid.getText().toString().trim().length()==0|| editName.getText().toString().trim().length()==0|| editsalary.getText().toString().trim().length()==0)

{

showMessage("Error", "Please enter all values"); return;

}

db.execSQL("INSERT INTO employee VALUES('"+editEmpid.getText()+"','"+editName.getText()+

"','"+editsalary.getText()+"');"); showMessage("Success", "Record added"); clearText();

}

if(view==btnDelete)

{

if(editEmpid.getText().toString().trim().length()==0)

{

showMessage("Error", "Please enter Employee id"); return;

}

Cursor c=db.rawQuery("SELECT \* FROM employee WHERE empid='"+editEmpid.getText()+"'", null); if(c.moveToFirst())

{

db.execSQL("DELETE FROM employee WHERE empid='"+editEmpid.getText()+"'"); showMessage("Success", "Record Deleted");

}

else

{

showMessage("Error", "Invalid Employee id");

}

clearText();

}

if(view==btnModify)

{

if(editEmpid.getText().toString().trim().length()==0)

{

showMessage("Error", "Please enter Employee id"); return;

}

Cursor c=db.rawQuery("SELECT \* FROM employee WHERE empid='"+editEmpid.getText()+"'", null); if(c.moveToFirst())

{

db.execSQL("UPDATE employee SET name='"+editName.getText()+"',salary='"+editsalary.getText()+ "' WHERE empid='"+editEmpid.getText()+"'"); showMessage("Success", "Record Modified");

}

else

{

showMessage("Error", "Invalid Rollno");

}

clearText();

}

if(view==btnView)

{

if(editEmpid.getText().toString().trim().length()==0)

{

showMessage("Error", "Please enter Employee id"); return;

}

Cursor c=db.rawQuery("SELECT \* FROM employee WHERE empid='"+editEmpid.getText()+"'", null); if(c.moveToFirst())

{

editName.setText(c.getString(1)); editsalary.setText(c.getString(2));

}

else

{

showMessage("Error", "Invalid Employee id"); clearText();

}

}

if(view==btnViewAll)

{

Cursor c=db.rawQuery("SELECT \* FROM employee", null); if(c.getCount()==0)

{

showMessage("Error", "No records found"); return;

}

StringBuffer buffer=new StringBuffer(); while(c.moveToNext())

{

buffer.append("Employee id: "+c.getString(0)+"\n"); buffer.append("Name: "+c.getString(1)+"\n"); buffer.append("salary: "+c.getString(2)+"\n\n");

}

showMessage("Employee details 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()

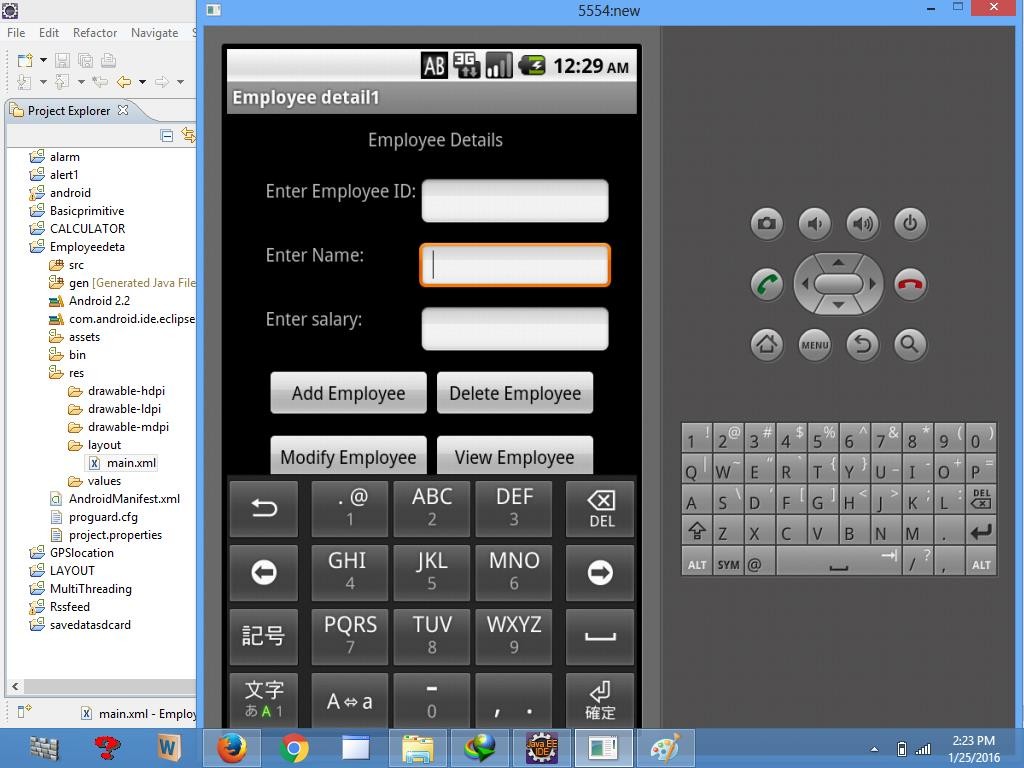
{

editEmpid.setText(""); editName.setText(""); editsalary.setText(""); editEmpid.requestFocus();

}

}

**Output**:



# Practical No.-6

### Aim: Implement an application that implements Multithreading.

**Hardware Requirements & Software Requirements:**

* Android Studio3.0
* Windows 7/8/10 OS
* RAM 4 GB
* Hard disk 10 GB,
* Emulator/mobile phones etc;

### Procedure:

1. Open android studio and select new android project.
2. Give project name and select next.
3. Choose the minimum target API version and select next.
4. Enter the package name, and its domain. Package name must be of two words separated by comma and then click on finish.
5. Go to package explorer in the left-hand side and select your project.
6. Go to res folder and select layout. Double click the main.xml file.
7. Now you can see the Graphics layout window.

**Source code:**

### Front-end:

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

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

### Backend:

package multi.threading;

//import your.first.R; import android.app.Activity; import android.os.Bundle; import android.os.Handler; import android.view.View;

import android.widget.TextView;

public class MultiThreadingActivity 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.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");

}

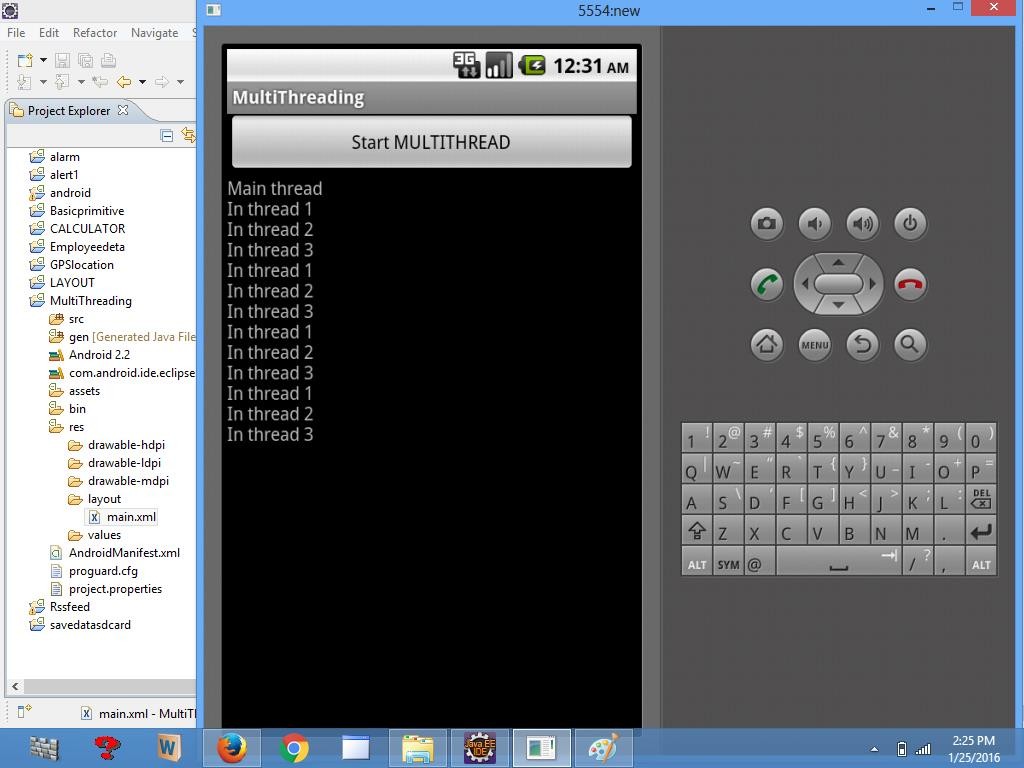
if(msg.what == t3) { tvOutput.append("\nIn thread 3");

}

} };

}

**Output:**



# Practical No.-7

### Aim: Develop a native application that uses GPS location information.

**Hardware Requirements & Software Requirements:**

* Android Studio3.0
* Windows 7/8/10 OS
* RAM 4 GB
* Hard disk 10 GB,
* Emulator/mobile phones etc;

### Procedure:

1. Open android studio and select new android project.
2. Give project name and select next.
3. Choose the minimum target API version and select next.
4. Enter the package name, and its domain. Package name must be of two words separated by comma and then click on finish.
5. Go to package explorer in the left-hand side and select your project.
6. Go to res folder and select layout. Double click the main.xml file.
7. Now you can see the Graphics layout window.

**Source code:**

### Front-end:

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

<RelativeLayout [xmlns:android="http://schemas.android.com/apk/res/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/show\_Location" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content android:text="Show\_Location" android:layout\_centerVertical="true"

android:layout\_centerHorizontal="true"

/>

</RelativeLayout>

### Backend:

package gps.location;

//import android.R;

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

public class GPSlocationActivity 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.main); btnShowLocation=(Button)findViewById(R.id.show\_Location); btnShowLocation.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

// TODO Auto-generated method stub

gps=new GPStrace(GPSlocationActivity.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();

}

} }); } })

package gps.location;

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.G PS\_PROVIDER);

isNetworkEnabled=locationManager.isProviderEnabled(LocationManag er.NETWORK\_PROVI

DER);

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.NE TWORK\_PROVIDER)

;

if(location !=null){ latitude=location.getLatitude(); longtitude=location.getLongitude();

}}}

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

locationManager.requestLocationUpdates(LocationManager.GPS\_PROVI DER,MIN\_TIME\_B

W\_UPDATES, MIN\_DISTANCE\_CHANGE\_FOR\_UPDATES, this);

if(locationManager!=null){ location=locationManager.getLastKnownLocation(LocationManager.GP S\_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;

}}

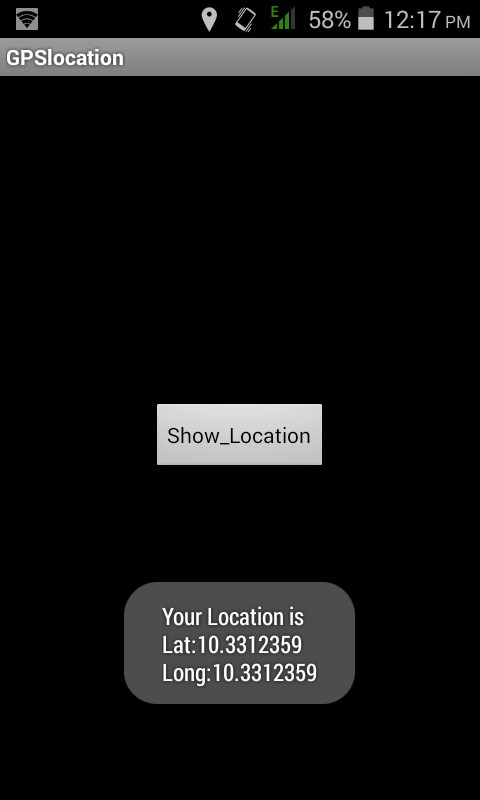
1. Go to manifest.xml file and add the code below

<uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION"/>

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

* + Now go to main.xml and right click.select run as option and select run configuration
  + Android output is present in the android emulator as shown in below.

**Output:**



# Practical No.-8

### Aim: Implement an application that creates an alert upon receiving a message.

**Hardware Requirements & Software Requirements:**

* Android Studio3.0
* Windows 7/8/10 OS
* RAM 4 GB
* Hard disk 10 GB,
* Emulator/mobile phones etc;

### Procedure:

1. Open android studio and select new android project.
2. Give project name and select next.
3. Choose the minimum target API version and select next.
4. Enter the package name, and its domain. Package name must be of two words separated by comma and then click on finish.
5. Go to package explorer in the left-hand side and select your project.
6. Go to res folder and select layout. Double click the main.xml file.
7. Now you can see the Graphics layout window.

**Source code:**

### Front-end:

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

<ScrollView [xmlns:android="http://schemas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android) android:layout\_width="fill\_parent" android:layout\_height="wrap\_content" android:scrollbars="vertical" >

<TableLayout android:layout\_width="match\_parent" android:layout\_height="wrap\_content"

android:shrinkColumns="\*" android:stretchColumns="\*" android:background="#000000">

<TableRow android:layout\_height="wrap\_content" android:layout\_width="match\_parent" android:gravity="center\_horizontal">

<TextView android:id="@+id/Title"

android:layout\_width="fill\_parent" android:layout\_height="wrap\_content" android:layout\_margin="5px" android:focusable="false" android:focusableInTouchMode="false" android:gravity="center\_vertical|center\_horizontal" android:text="QUIZ"

android:textSize="25sp" android:textStyle="bold" />

<View android:layout\_height="2px" android:layout\_marginTop="5dip"

android:layout\_marginBottom="5dip" android:background="#DDFFDD"/>

</TableRow>

<TableRow android:layout\_height="wrap\_content" android:layout\_width="match\_parent" android:gravity="center\_horizontal">

<TextView android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:textSize="18sp" android:text="1.CAPTIAL OF INDIA" android:layout\_span="4" android:padding="18dip" android:textColor="#ffffff"/>

</TableRow>

<TableRow android:id="@+id/tableRow1" android:layout\_height="wrap\_content" android:layout\_width="match\_parent">

<RadioGroup android:id="@+id/answer1" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:layout\_weight="0.4" >

<RadioButton android:id="@+id/answer1A" android:layout\_width="match\_parent" android:layout\_height="wrap\_content"

android:textColor="#ffffff" android:text="CHENNAI" />

<RadioButton android:id="@+id/answer1B" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:textColor="#ffffff" android:text="NEW DELHI" />

<RadioButton android:id="@+id/answer1C" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:textColor="#ffffff" android:text="MUMBAI" />

<RadioButton android:id="@+id/answer1D" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:textColor="#ffffff" android:text="HYDERBAD" />

</RadioGroup>

</TableRow>

<TableRow android:layout\_height="wrap\_content" android:layout\_width="match\_parent" android:gravity="center\_horizontal">

<TextView android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:textSize="18sp"

android:text="2. CAPTIAL OF RUSSIA?" android:layout\_span="4" android:padding="18dip"

android:textColor="#ffffff"/>

</TableRow>

<TableRow android:id="@+id/tableRow2" android:layout\_height="wrap\_content" android:layout\_width="match\_parent">

<RadioGroup android:id="@+id/answer2" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:layout\_weight="0.4" >

<RadioButton android:id="@+id/answer2A" android:layout\_width="match\_parent" android:layout\_height="wrap\_content"

android:textColor="#ffffff" android:text="WARSAW " />

<RadioButton android:id="@+id/answer2B" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:textColor="#ffffff" android:text="BERLIN" />

<RadioButton android:id="@+id/answer2C" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:textColor="#ffffff" android:text="MASCOW " />

<RadioButton android:id="@+id/answer2D" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:textColor="#ffffff" android:text="CANEBRA " />

</RadioGroup>

</TableRow>

<TableRow

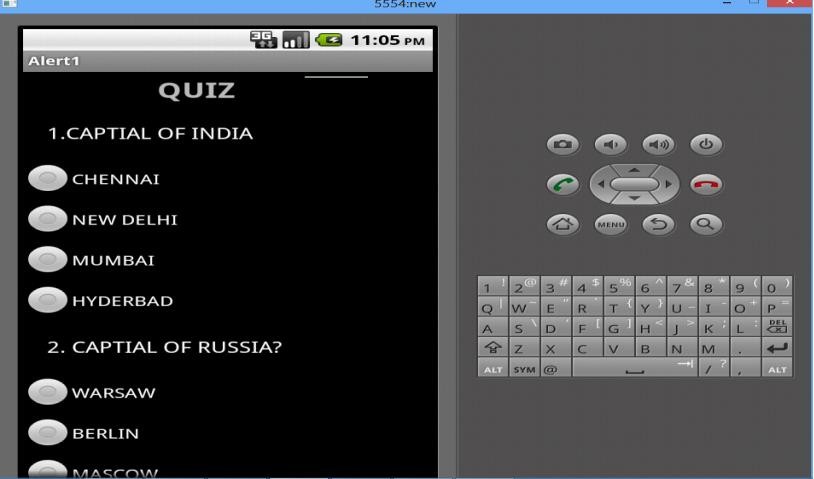
<Button android:id="@+id/submit"

android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:gravity="center" android:text="Submit" />

</TableRow>

</TableLayout>

</ScrollView>

**Output:**

# Practical No.-9

### Aim: Write a mobile application that creates alarm clock.

**Hardware Requirements & Software Requirements:**

* Android Studio3.0
* Windows 7/8/10 OS
* RAM 4 GB
* Hard disk 10 GB,
* Emulator/mobile phones etc;

### Procedure:

1. Open android studio and select new android project.
2. Give project name and select next.
3. Choose the minimum target API version and select next.
4. Enter the package name, and its domain. Package name must be of two words separated by comma and then click on finish.
5. Go to package explorer in the left-hand side and select your project.
6. Go to res folder and select layout. Double click the main.xml file.
7. Now you can see the Graphics layout window.

**Source code:**

### Front-end:

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

<manifest [xmlns:android="http://schemas.android.com/apk/res/and](http://schemas.android.com/apk/res/android)roid" package="com.javapapers.androidalarmclock">

<uses-permission android:name="android.permission.WAKE\_LOCK" />

<application android:allowBackup="true" android:icon="@drawable/ic\_launcher" android:label="@string/app\_name" android:theme="@style/AppTheme">

<activity

android:name=".AlarmActivity" android:label="@string/app\_name">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

<service android:name=".AlarmService" android:enabled="true" />

<receiver android:name=".AlarmReceiver" />

</application>

</manifest>

**//**\*Android Activity\***//**

//\*activity\_my.xml**\*//**

//\*The Android Activity is designed to be simple. We have a TimePicker component followed by a ToggleButton. That’s it. Choose the time to set the alarm and toggle the switch to on. The alarm will work\*//

<RelativeLayout [xmlns:android="http://schemas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android) [xmlns:tools="http://schemas.android.com/tools"](http://schemas.android.com/tools) android:layout\_width="match\_parent" android:layout\_height="match\_parent" android:paddingLeft="@dimen/activity\_horizontal\_margin" android:paddingRight="@dimen/activity\_horizontal\_margin" android:paddingTop="@dimen/activity\_vertical\_margin" android:paddingBottom="@dimen/activity\_vertical\_margin" tools:context=".MyActivity">

<TimePicker android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:id="@+id/alarmTimePicker" android:layout\_alignParentTop="true" android:layout\_centerHorizontal="true" />

<ToggleButton android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="Alarm On/Off" android:id="@+id/alarmToggle" android:layout\_centerHorizontal="true" android:layout\_below="@+id/alarmTimePicker" android:onClick="onToggleClicked" />

<TextView

android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:textAppearance="?android:attr/textAppearanceLarge" android:text=""

android:id="@+id/alarmText" android:layout\_alignParentBottom="true" android:layout\_centerHorizontal="true" android:layout\_marginTop="20dp" android:layout\_below="@+id/alarmToggle" />

</RelativeLayout>

## Backend:

**//**\*AlarmActivity.java\*//

AlarmActivity uses the AlarmManager to set the alarm and send notification on alarm trigger.

package com.javapapers.androidalarmclock; import android.app.Activity;

import android.app.AlarmManager; import android.app.PendingIntent; import android.content.Intent; import android.os.Bundle;

import android.util.Log; import android.view.View;

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

public class AlarmActivity extends Activity { AlarmManager alarmManager;

private PendingIntent pendingIntent; private TimePicker alarmTimePicker; private static AlarmActivity inst; private TextView alarmTextView;

public static AlarmActivity instance() { return inst;

}

@Override

public void onStart() { super.onStart();

inst = this;

}

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_my);

alarmTimePicker = (TimePicker) findViewById(R.id.alarmTimePicker);

alarmTextView = (TextView) findViewById(R.id.alarmText); ToggleButton alarmToggle = (ToggleButton) findViewById(R.id.alarmToggle);

alarmManager = (AlarmManager) getSystemService(ALARM\_SERVICE);

}

public void onToggleClicked(View view) { if (((ToggleButton) view).isChecked()) { Log.d("MyActivity", "Alarm On");

Calendar calendar = Calendar.getInstance(); calendar.set(Calendar.HOUR\_OF\_DAY, alarmTimePicker.getCurrentHour()); calendar.set(Calendar.MINUTE, alarmTimePicker.getCurrentMinute());

Intent myIntent = new Intent(AlarmActivity.this, AlarmReceiver.class);

pendingIntent = PendingIntent.getBroadcast(AlarmActivity.this, 0,

myIntent, 0);

alarmManager.set(AlarmManager.RTC, calendar.getTimeInMillis(), pendingIntent);

} else { alarmManager.cancel(pendingIntent); setAlarmText(""); Log.d("MyActivity", "Alarm Off");

}

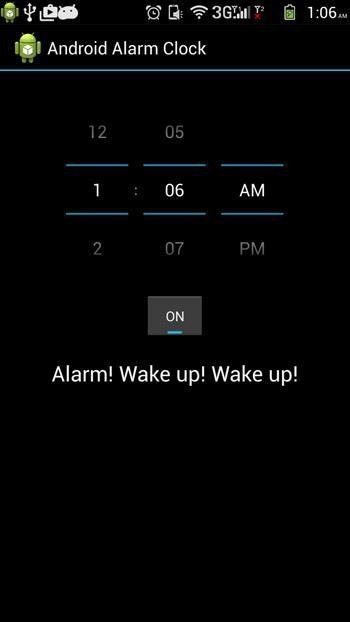
}

public void setAlarmText(String alarmText) { alarmTextView.setText(alarmText);

}

}

**Output:**



# Practical No.-10

### Aim: Implement an application that writes data to the SD card.

**Hardware Requirements & Software Requirements:**

* Android Studio3.0
* Windows 7/8/10 OS
* RAM 4 GB
* Hard disk 10 GB,
* Emulator/mobile phones etc;

### Procedure:

1. Open android studio and select new android project.
2. Give project name and select next.
3. Choose the minimum target API version and select next.
4. Enter the package name, and its domain. Package name must be of two words separated by comma and then click on finish.
5. Go to package explorer in the left-hand side and select your project.
6. Go to res folder and select layout. Double click the main.xml file.
7. Now you can see the Graphics layout window.

**Source code:**

### Front-end:

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

<LinearLayout [xmlns:android="http://schemas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android) android:layout\_width="fill\_parent" android:layout\_height="fill\_parent" android:background="#ff0000ff" android:orientation="vertical" >

<EditText android:id="@+id/editText1" android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" >

<requestFocus />

</EditText>

<Button android:id="@+id/button1"

android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:text="SAVE DATA" />

<Button android:id="@+id/button2"

android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:text="SHOW DATA" />

<TextView android:id="@+id/textView1" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"

/>

</LinearLayout>

## Backend:

package save.sd; import java.io.File;

import java.io.FileInputStream; import java.io.FileNotFoundException; import java.io.FileOutputStream; import java.io.IOException;

import java.io.InputStreamReader; import java.io.OutputStreamWriter; import android.app.Activity; import android.os.Bundle;

import android.os.Environment; import android.view.View; import android.widget.Button; import android.widget.EditText; import android.widget.TextView; import android.widget.Toast;

public class SavedatasdcardActivity extends Activity {

/\*\* Called when the activity is first created. \*/ Button save,load;

EditText message; TextView t1; String Message1; @Override

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

save=(Button) findViewById(R.id.button1); load=(Button) findViewById(R.id.button2); message=(EditText) findViewById(R.id.editText1); t1=(TextView) findViewById(R.id.textView1); save.setOnClickListener(new View.OnClickListener(){ public void onClick(View v){

//Get message from user store in message1 variable Message1 =message.getText().toString();

try{

//Create a new folder called MyDirectory in SDCard File sdcard=Environment.getExternalStorageDirectory(); File directory=new File(sdcard.getAbsolutePath()+"/MyDirectory"); directory.mkdirs();

//Create a new file name textfile.txt inside MyDirectory File file=new File(directory,"textfile.txt");

//Create File Outputstream to read the file FileOutputStream fou=new FileOutputStream(file); OutputStreamWriter osw=new OutputStreamWriter(fou); try{

//write a user data to file osw.append(Message1); osw.flush();

osw.close(); Toast.makeText(getBaseContext(),"Data Saved",Toast.LENGTH\_LONG).show();

}catch(IOException e){ e.printStackTrace();

}

}catch (FileNotFoundException e){ e.printStackTrace();

}

}

});

load.setOnClickListener(new View.OnClickListener(){ public void onClick(View v){

try{

File sdcard=Environment.getExternalStorageDirectory(); File directory=new File(sdcard.getAbsolutePath()+"/MyDirectory");

File file=new File(directory,"textfile.txt"); FileInputStream fis=new FileInputStream(file); InputStreamReader isr=new InputStreamReader(fis); char[] data=new char[100];

String final\_data=""; int size;

try{

while((size=isr.read(data))>0)

{

//read a data from file

String read\_data=String.copyValueOf(data,0,size); final\_data+=read\_data;

data=new char[100];

}

//display the data in output Toast.makeText(getBaseContext(),"Message:"+final\_data,Toast.LENG TH\_LONG).show();

}catch(IOException e){ e.printStackTrace();

}}catch (FileNotFoundException e){ e.printStackTrace();

}}});}}

### C:\Users\CCET\Desktop\ss\9.pngOutput:

**https://github.com/karanabrol15/MADLAB.git**