Vel Tech Multi Tech

Dr.Rangarajan Dr.Sakunthala Engineering College

An Autonomous Institution

Approved by AICTE, Affiliated to Anna University, Chennai.
ISO 9001:2015 Certified Institution, Accredited by NBA (BME, CSE, ECE, EEE, IT & MECH),
Accredited by NAAC with 'A' Grade with CGPA of 3.49.
#42, Avadi-Vel Tech Road, Avadi, Chennai- 600062, Tamil Nadu, India.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

191CS52A / MOBILE APPLICATION DEVELOPMENT LABORATORY

NAME :

REGISTERNO :

ROLLNO :

BRANCH :

YEAR :

SEMESTER :

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Vision

- To emerge as centre for academic excellence in the field of Computer Science and Engineering by exposure to research and industry practices

Mission

- To provide good teaching and learning environment with conductive research atmosphere in the field of Computer Science and Engineering.
- To propagate lifelong learning.
- To impart the right proportion of knowledge, attitudes and ethics in students to enable them take up positions of responsibility in the society and make significant contributions.

Vel Tech Multi Tech

Dr.Rangarajan Dr.Sakunthala Engineering College

An Autonomous Institution

Approved by AICTE, Affiliated to Anna University, Chennai.
ISO 9001:2015 Certified Institution, Accredited by NBA (BME, CSE, ECE, EEE, IT & MECH),
Accredited by NAAC with 'A' Grade with CGPA of 3.49.
#42, Avadi-Vel Tech Road, Avadi, Chennai- 600062, Tamil Nadu, India.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

NAME: …….. YEAR: ……SEMESTER: …..…. ..BRANCH: ……

University Register No: | | COLLEGE ROLL NO: …….

Certified that, this is the bonafide record of work done by the above student in **191CS52A - MOBILE APPLICATION DEVELOPMENT LABORATORY** during the academic year 2024 – 2025.

Signature of Head of the Department **Signature of Staff In-charge**

Submitted for the University Practical Examination held on..... at

VELTECH MULTITECH Dr RANGARAJAN Dr SAKUNTHALA ENGINEERING COLLEGE,

N0.42, AVADI-VEL TECH ROAD, CHENNAI- 600 062.

Signature of Examiner's

INTERNAL EXAMINER **EXTERNAL EXAMINER**

Department of Computer Science and Engineering

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

PEOs	Programme Educational Objectives (PEOs)
PEO1	Ability to identify, formulate and analyze complex Computer Science and Engineering problems in the areas of hardware, software, theoretical Computer Science and applications to reach significant conclusions by applying Mathematics, Natural sciences, Computer Science and Engineering principles.
PEO2	Apply knowledge of mathematics, natural science, engineering fundamentals and system fundamentals, software development, networking & communication, and information security to the solution of complex engineering problems in computer science and engineering to get benefits in their professional career or higher education and research or technological entrepreneur.
PEO3	Design solutions for complex computer science and engineering problems using state of the art tools and techniques, components or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.

PROGRAMME SPECIFIC OUTCOMES (PSOs)

PSO's	PROGRAMME SPECIFIC OUTCOMES (PSOs)
PSO1	An ability to apply, design and development of application oriented software systems and to test and document in accordance with Computer Science and Engineering.
PSO2	The design techniques, analysis and the building, testing, operation and maintenance of networks, databases, security and computer systems (both hardware and software).
PSO3	An ability to identify, formulate and solve hardware and software problems using sound computer engineering principles.

Department of Computer Science and Engineering

PROGRAMME OUTCOMES (POs)

POs	Programme Outcomes (POs)
PO1	Apply knowledge of **mathematics, natural science**, **engineering** fundamentals and system fundamentals, software development, networking & communication, and information assurance & security to the solution of complex engineering problems in computer science and engineering.
PO2	Ability to **identify**, **formulate** and **analyze complex** Computer Science and Engineering problems in the areas of hardware, software, theoretical Computer Science and applications to reach significant conclusions by applying Mathematics, Natural sciences, Computer Science and Engineering principles.
PO3	**Design solutions** for complex computer science and engineering problems and design systems, components or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.
PO4	Ability to use **research based knowledge** and **research methods** to perform literature survey, design experiments for complex problems in designing, developing and maintaining a computing system, collect data from the experimental outcome, analyze and interpret valid/interesting patterns and conclusions from the data points.
PO5	Ability to create, select and apply **state of the art tools** and techniques in designing, developing and testing a computing system or its component.
PO6	Apply reasoning informed by contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to **professional engineering practice** in system development and solutions to **complex engineering problems** related to system fundamentals, software development, networking & communication, and information assurance & security.
PO7	Understand and evaluate the sustainability and **impact of professional engineering** work in the solution of complex engineering problems related to system fundamentals, software development, networking & communication, and information assurance & security in societal and environmental contexts.
PO8	Apply **ethical principles** and commit to **professional ethics** and **responsibilities** and norms of computer science and engineering practice.
PO9	Ability to function as an individual and as a team player or leader in multidisciplinary teams and strive towards **achieving a common goal**.
PO10	**Communicate effectively** on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	Demonstrate knowledge and understanding of engineering management principles and economic decision making and apply these to one's own work, as a member and leader in a team, to **manage projects** and in multidisciplinary environments.
PO12	Recognize the need for, and have the preparation and ability to **engage in independent** and **lifelong learning** in the broadest context of technological change.

COURSE OBJECTIVES

- ✓ Build the components and structure of mobile application development frameworks for Android and windows OS based mobiles.
- ✓ Organize how to work with various mobile application development frameworks.
- ✓ Experiment the basic and important design concepts and issues of development of mobile applications.
- ✓ Demonstrate the capabilities and limitations of mobile devices

COURSE OUTCOMES

On completion of the course, students will be able to

CO1	Design the mobile applications using GUI and Layouts.
CO2	Appraise mobile applications using Event Listener.
CO3	Practice the mobile applications using Databases.
CO4	Apply mobile applications using RSS Feed, Internal/External Storage, SMS, Multithreading and GPS.
CO5	Create the own mobile app for simple needs.

***NOTE: 1 –Slight(Low) 2 –Moderate(Medium) 3 – Substantial(High)**

Course Outcome	Mapping CO's with PO's and PSO's														
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	3	3	3	1	-	-	-	-	-	-	3	2	-
CO2	3	3	3	3	3	1	-	-	-	-	-	-	3	2	-
CO3	3	3	3	3	3	3	-	-	-	-	-	-	3	2	-
CO4	3	3	3	3	3	1	-	-	-	-	-	-	3	2	-
CO5	3	3	3	3	3	1	-	-	-	-	-	-	3	2	-
CO	3	3	3	3	3	3	-	-	-	-	-	-	3	2	-

INDEX

S.NO	DATE	LIST OF EXPERIMENTS	CO	MARKS	PAGE NO	FACULTY SIGN
1		Develop an application that uses GUI components, Font and Colours	CO1			
2		Develop an application that uses Layout Managers and event listeners.	CO1,CO2			
3		Write an application that draws basic graphical primitives on the screen.	CO2			
4		Develop an application that makes use of databases.	CO3			
5		Develop an application that makes use of Notification Manager	CO3			
6		Implement an application that uses Multi-threading	CO4			
7		Develop a native application that uses GPS location information	CO4			
8		Implement an application that writes data to the SD card.	CO4			
9		Implement an application that creates an alert upon receiving a message	CO4			
10		Write a mobile application that makes use of RSS feed	CO4			
11		Develop a mobile application to send an email.	CO5			
12		Develop a Mobile application for simple needs (Mini Project)	CO5			

EX NO:1 **DATE:**

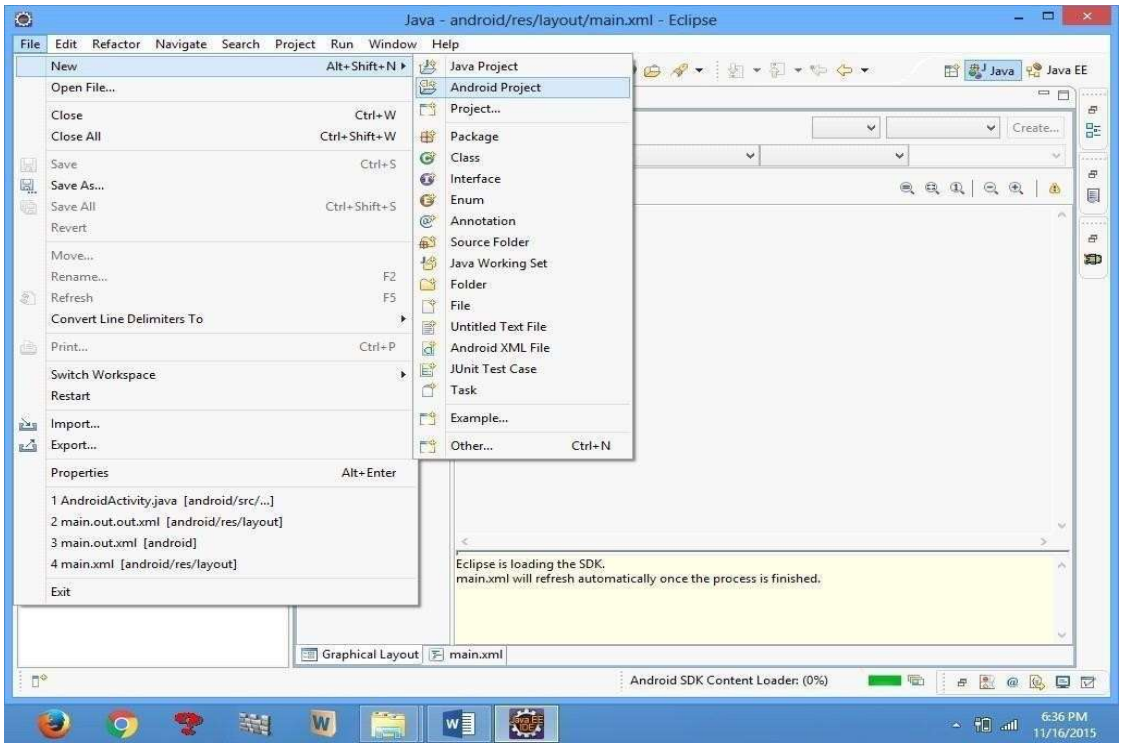
DEVELOP AN APPLICATION THAT USES GUI COMPONENTS, FONT AND COLOURS

AIM:

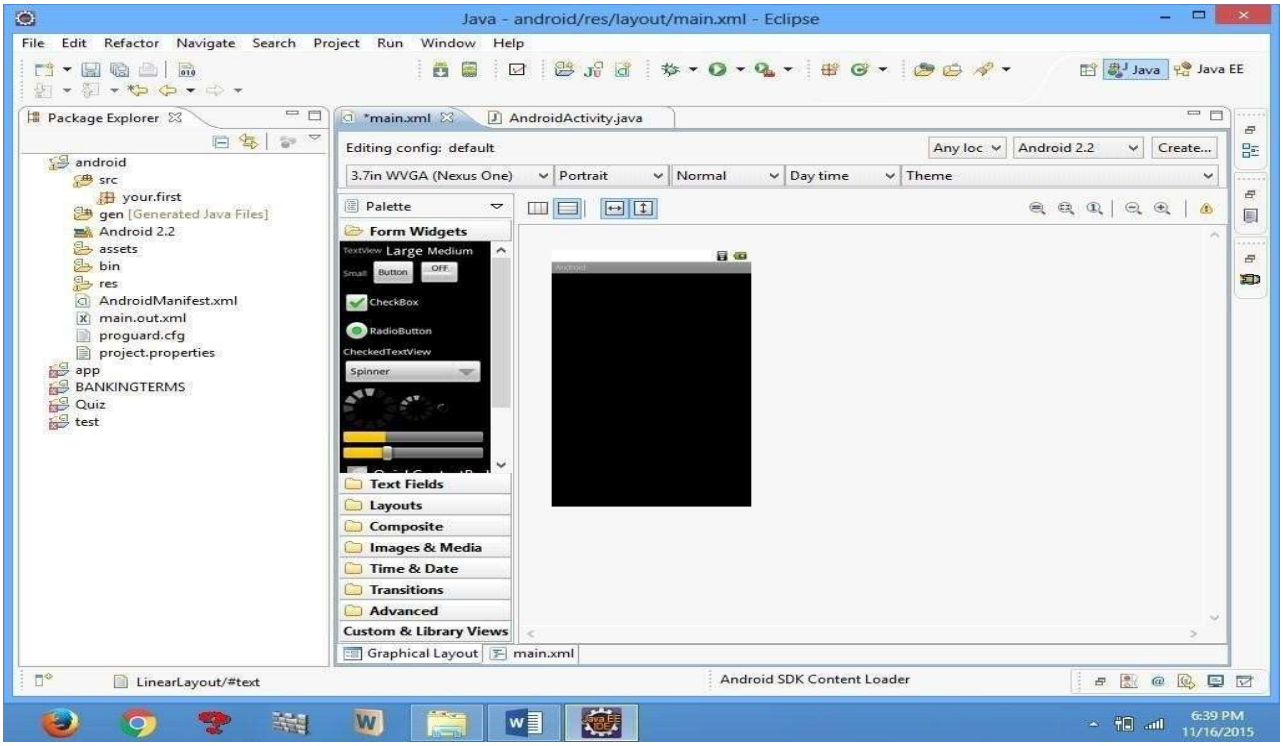
To develop an application that uses GUI components, Font and Colours.

PROCEDURE:

1. Open eclipse or android studio and select new android project.



2. Give project name and select next.
3. Choose the android version. Choose the lowest android version (Android 2.2) and select next.
4. Enter the package name. Package name must be two word separated by comma and click finish.
5. Go to package explorer in the left hand side. Select our project.
6. Go to res folder and select layout. Double click the main.xml file
7. Now you can see the Graphics layout window.

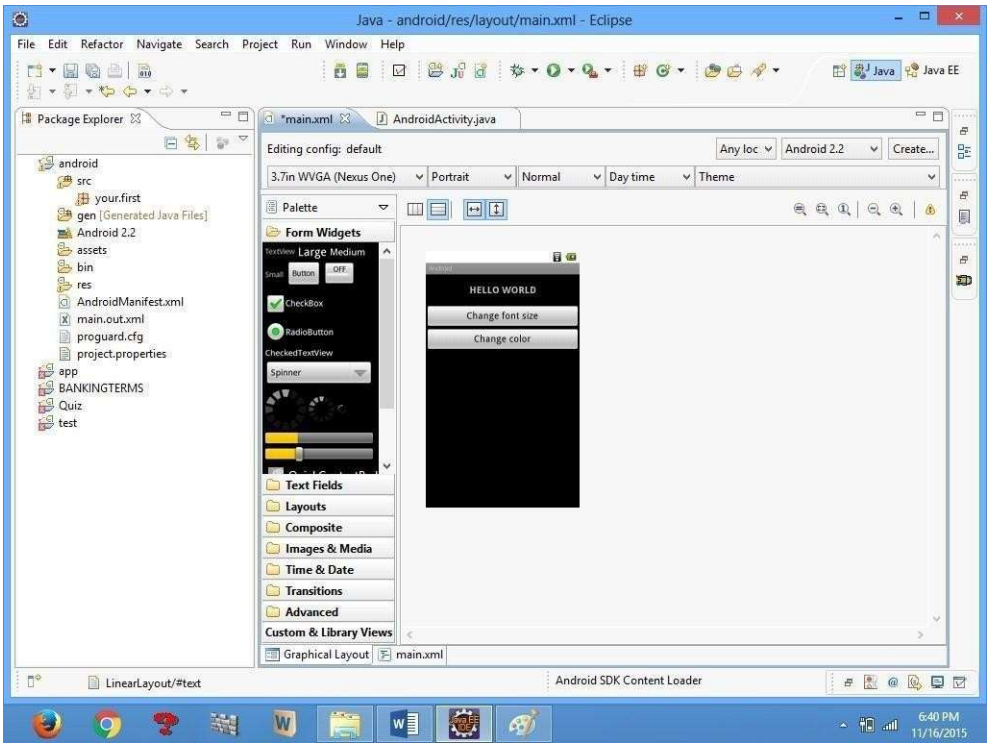


8) Click the main.xml file and type the code below

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

9) Again click the graphics layout tab and screen layout is look like below



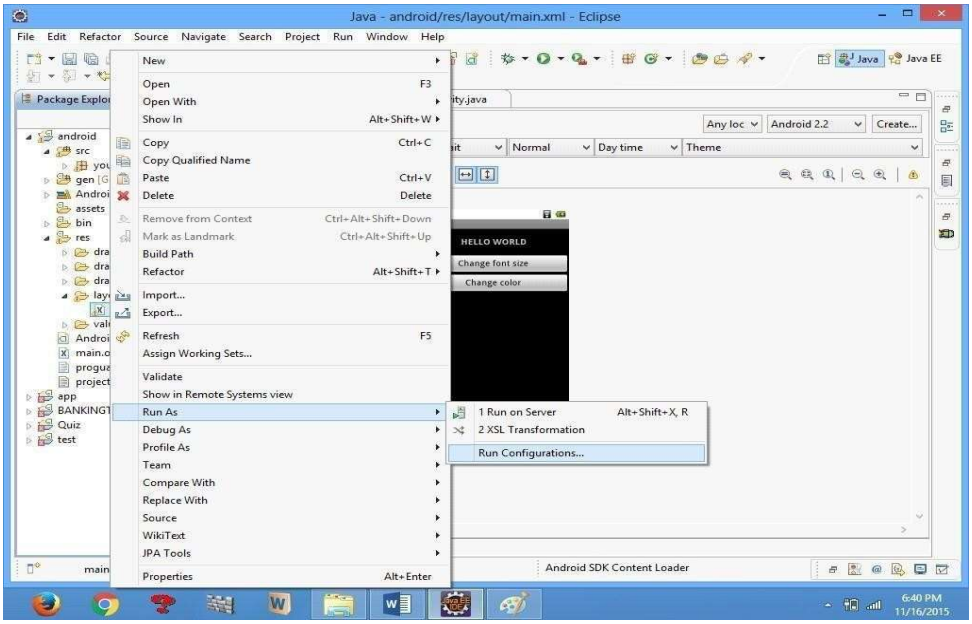
10) Go to project explorer and select src folder. Now select mainactivity.java file and type the following code

```
import android.R;
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;
}
});
}
}
```

11) Now go to main.xml and right click .select run as option and select run configuration



12) Android output is present in the android emulator as shown in below.

RESULT:

Thus an android application that uses GUI components, Font and Colours has been developedand executed successfully.

EX NO: 2 **DATE :**

DEVELOP AN APPLICATION THAT USES LAYOUT MANAGERS AND EVENT LISTENERS.

AIM:

To develop an application that uses layout managers and event listeners.

PROCEDURE:

1) Open eclipse or android studio and select new android project.
2) Give project name and select next.
3) Choose the android version. Choose the lowest android version (Android 2.2) and select next.
4) Enter the package name. Package name must be two word separated by comma and click finish.
5) Go to package explorer in the left hand side. Select our project.
6) Go to res folder and select layout. Double click the main.xml file. Add the code below.

```
<RelativeLayout xmlns: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>
```

7) Now select mainactivity.java file and type the following code.

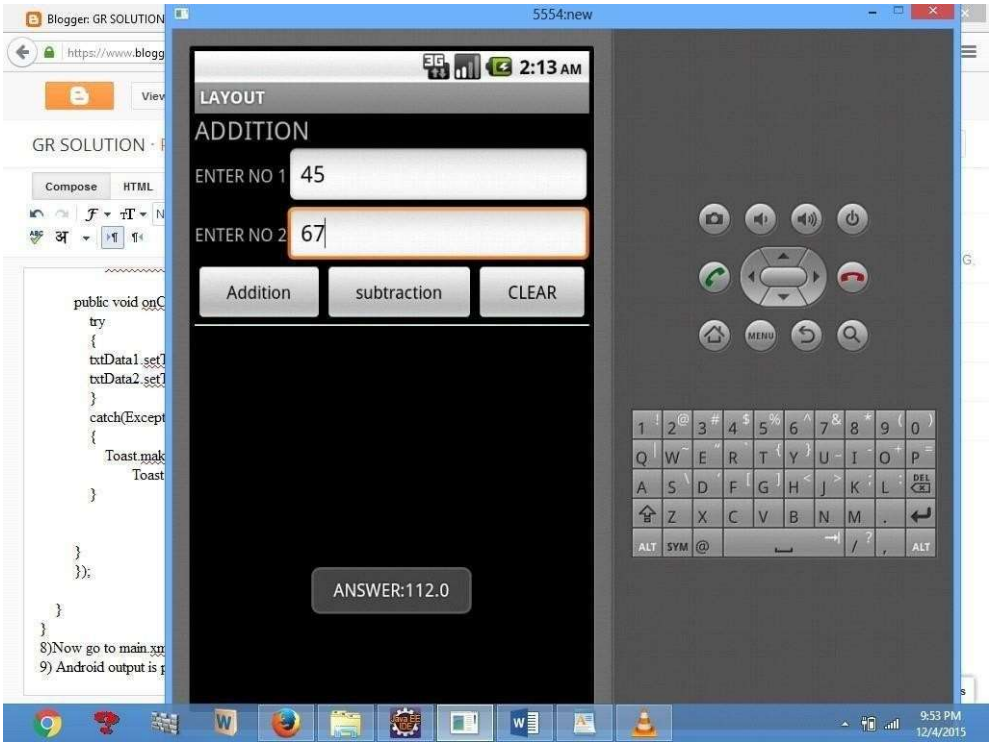
```
package layout.ne;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
public class LAYOUTActivity extends Activity {
/** Called when the 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_SHORT).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_SHORT).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();
}
}
});
}
}
```

8) Now go to main.xml and right click .select run as option and select run configuration.

9) Android output is present in the android emulator as shown in below.



RESULT:

Thus an android application that uses layout managers and event listeners has been developedand executed successfully.

EX NO: 3 **DATE:**

WRITE AN APPLICATION THAT DRAWS BASIC GRAPHICAL PRIMITIVES ON THE SCREEN IN ANDROID

AIM:

To develope an application that draws basic graphical primitives on the screen in android.

PROCEDURE:

1) Open eclipse or android studio and select new android project.
2) Give project name and select next.
3) Choose the android version. Choose the lowest android version (Android 2.2) and select next.
4) Enter the package name. Package name must be two word separated by comma and click finish.
5) Go to package explorer in the left hand side. Select our project.
6) Go to res folder and select layout. Double click the main.xml file. Don't change anything in layout. Leave as default.
7) Now select mainactivity.java file and type the following code.

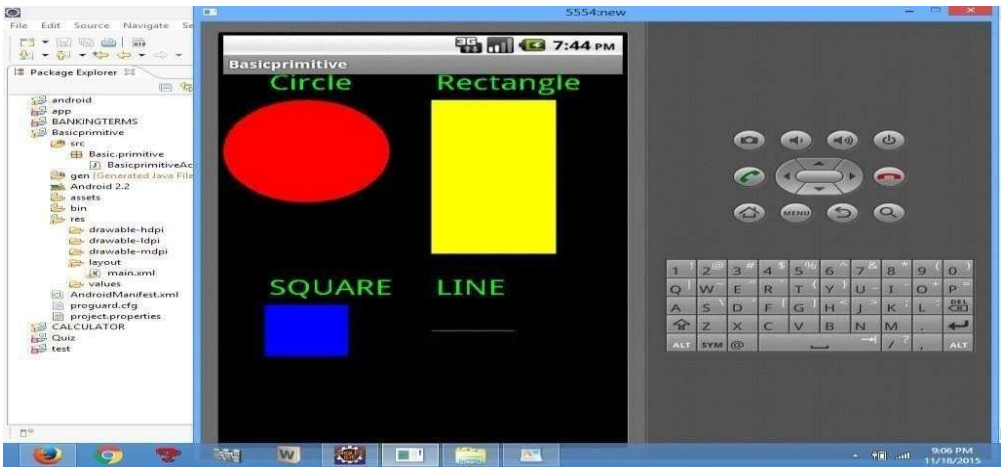
```
package Basic.primitive;
import android.app.Activity;
 import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.os.Bundle;
import android.view.View;
public class BasicprimitiveActivity extends Activity {
/** Called 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);
}}
}
```

8) Now go to main.xml and right click .select run as option and select run configuration.

9) Android output is present in the android emulator as shown in below.

OUTPUT:



RESULT :

Thus an android application that draws basic graphical primitives on the screen in android has been developed and executed successfully.

EX NO: 4 **DATE:**

DEVELOP AN APPLICATION THAT MAKES USE OF DATABASE

AIM:

To develop an application that makes use of database.

PROCEDURE:

1) Open eclipse or android studio and select new android project.
2) Give project name and select next.
3) Choose the android version. Choose the lowest android version(Android 2.2) and select next
4) Enter the package name. Package name must be two word separated by comma and click finish.
5) Go to package explorer in the left hand side. Select our project.
6) Go to res folder and select layout. Double click the main.xml file. Add the code below

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

7) 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>
```

8) Now select mainactivity.java file and type the following code.In my coding mainactivity name is EmployeedetailActivity.

```
package student.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 Studentdetail1Activity 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");
}
```

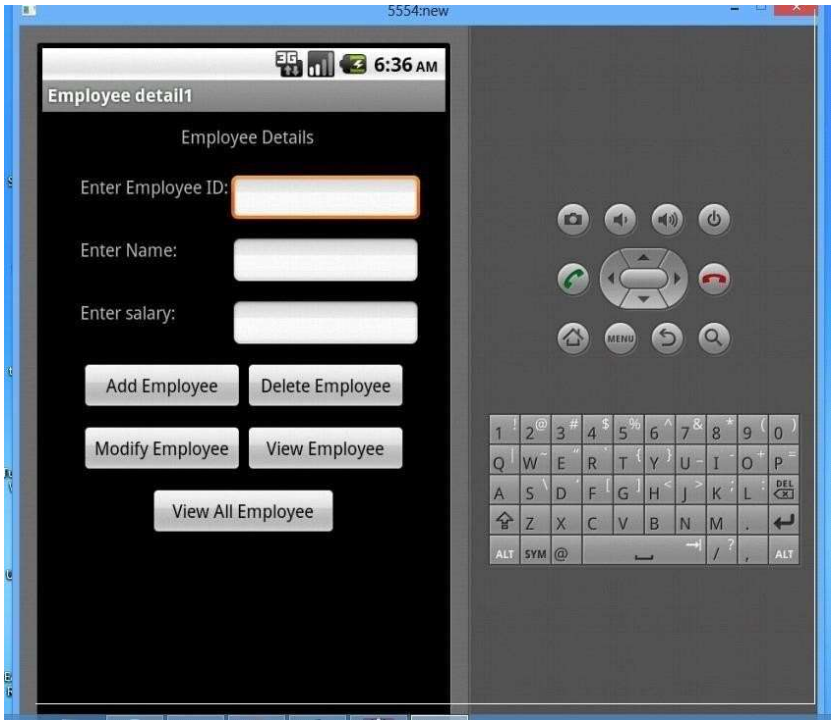
```
else
{
showMessage("Error", "Invalid Employee id");
}
clearText();
}
if(view==btnModify)
{
if(editEmpid.getText().toString().trim().length()==0)
{
showMessage("Error", "Please enter Employee id");
return;
}
Cursor c=db.rawQuery("SELECT * FROM employee WHERE
    empid='"+editEmpid.getText()+"'", null);
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();
}
}
```

9) Now go to main.xml and right click .select run as option and select run configuration.

10) Android output is present in the android emulator as shown in below.



RESULT:

Thus an application that makes use of database has been developed and executed successfully.

EX NO 5: **DATE :**

DEVELOP AN APPLICATION THAT MAKES USE OF NOTIFICATION MANAGER

AIM:

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

PROCEDURE:

1) Open eclipse or android studio and select new android project.
2) Give project name and select next.
3) Choose the android version. Choose the lowest android version (Android 2.2) and select next.
4) Enter the package name. Package name must be two word separated by comma and click finish.
5) Go to package explorer in the left hand side. Select our project.
6) Go to res folder and select layout. Double click the main.xml file. Add the code below.

```
<?xml version="1.0" encoding="utf-8"?>
    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:layout_margin="10dp"
      android:orientation="vertical">

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

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

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

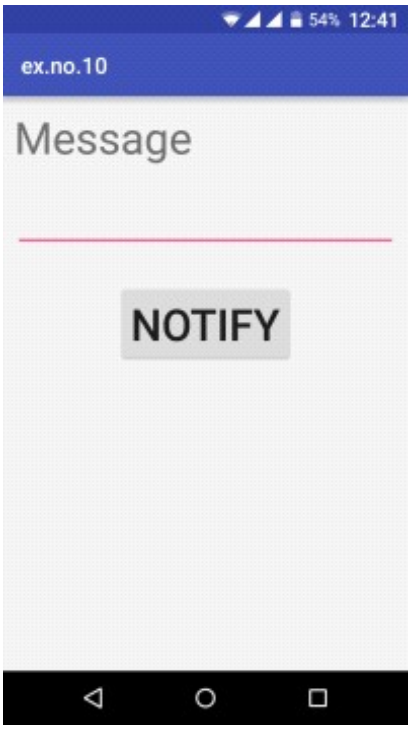
7) Now select mainactivity.java file and type the following code.In my coding mainactivity.

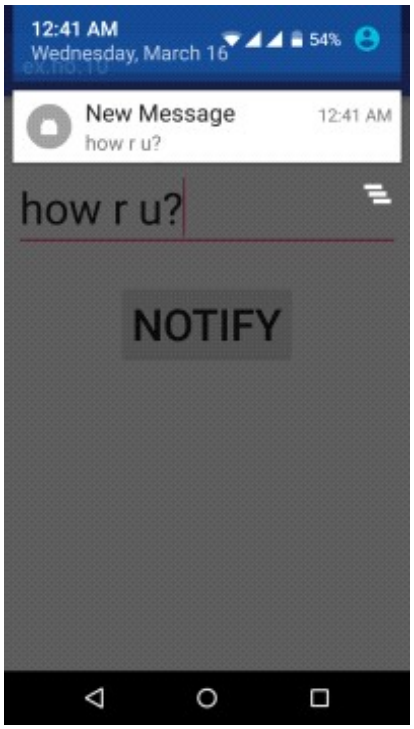
```
package com.example.exno10;
import android.app.Notification;
import android.app.NotificationManager;
import  android.app.PendingIntent;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
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)
              {0);
Intent intent = new Intent(MainActivity.this, SecondActivity.class); PendingIntent pending =
PendingIntent.getActivity(MainActivity.this, 0, intent,

Notification noti = new
        Notification.Builder(MainActivity.this).setContentTitle("New
        Message").setContentText(e.getText().toString()).setSmallIcon(R.mipmap.ic_launche
        r).setContentIntent(pending).build();
              NotificationManager manager = (NotificationManager)
        getSystemService(NOTIFICATION_SERVICE);
              noti.flags |= Notification.FLAG_AUTO_CANCEL;
              manager.notify(0, noti);
           }
        });
     }
  }
```

8) So now the Coding part is also completed.

9) Now run the application to see the output.





RESULT:

Thus an android Application that creates an alert upon receiving a message has been developedand executed successfully.

EX NO:6 **DATE :**

IMPLEMENT AN APPLICATION THAT IMPLEMENTS MULTI THREADING

AIM:

To implement an application that implements multithreading.

PROCEDURE:

1) Open eclipse or android studio and select new android project.
2) Give project name and select next.
3) Choose the android version. Choose the lowest android version (Android 2.2) and select next.
4) Enter the package name. Package name must be two word separated by comma and click finish.
5) Go to package explorer in the left hand side. Select our project.
6) Go to res folder and select layout. Double click the main.xml file. Add the code below.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
android:id="@+id/info" >
<Button
android:id="@+id/button1"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:onClick="fetchData"
android:text="Start MULTITHREAD" />
<TextView
android:id="@+id/textView1"
```

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Main thread" />
</LinearLayout>
```

7) Now select mainactivity.java file and type the following code.

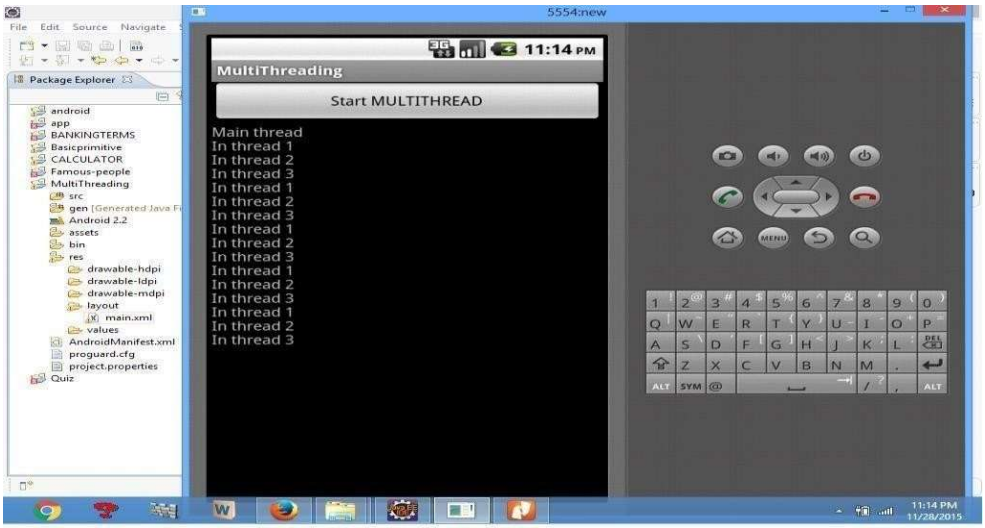
```
Ypackage multi.threading;
//import your.first.R;
import android.app.Activity;
import  android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.widget.TextView;
public class MultiThreadingActivity extends Activity {
private  TextView  tvOutput;
private static final int 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");
}
}
};
}
```

8) Now go to main.xml and right click .select run as option and select run configuration.

9) Android output is present in the android emulator as shown in below.



RESULT:

Thus an android application that implement multithreading has been developed and executed successfully.

EX NO:7 **DATE :**

DEVELOP A NATIVE APPLICATION THAT USES GPS LOCATION INFORMATION

AIM:

To develop a native application that uses GPS location information.

PROCEDURE:

1) Open eclipse or android studio and select new android project.
2) Give project name and select next.
3) Choose the android version. Choose the lowest android version (Android 2.2) and select next.
4) Enter the package name. Package name must be two word separated by comma and click finish.
5) Go to package explorer in the left hand side. Select our project.
6) Go to res folder and select layout. Double click the main.xml file. Add the code below

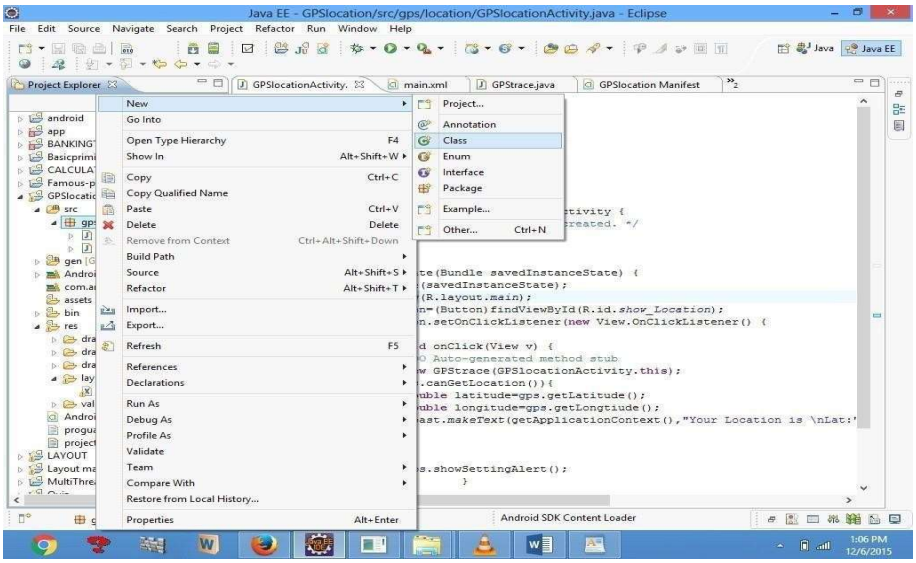
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:id="@+id/relativeLayout1"
android:layout_width="match_parent"
android:layout_height="match_parent" >
<Button
android:id="@+id/show_Location"
android:layout_width="wrap_content"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

7) Now select mainactivity.java file and type the following code.In my coding maniactivity name is GPSlocationActivity.

```
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();
}
}
});
}
}
```

8) Go to src folder and Right Click on your package folder and choose new class and give the class nams as GPStrace.



9) Select the GPStrace.java file and paste the following code.

```
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.GPS_PROVIDER);
   isNetworkEnabled=locationManager.isProviderEnabled(LocationManager.NETWORK_
     PROVIDER);
if(!isGPSEnabled && !isNetworkEnabled){
}else{
this.canGetLocation=true;
if(isNetworkEnabled){
locationManager.requestLocationUpdates(
LocationManager.NETWORK_PROVIDER,
MIN_TIME_BW_UPDATES,
MIN_DISTANCE_CHANGE_FOR_UPDATES,this);
}
if(locationManager!=null){
     location=locationManager.getLastKnownLocation(LocationManager.NETWORK_PRO
     VIDER);
if(location !=null){
latitude=location.getLatitude();
longtitude=location.getLongitude();
}
}
}
if(isGPSEnabled){
if(location==null){
     locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,MIN_TI
     ME_BW_UPDATES, MIN_DISTANCE_CHANGE_FOR_UPDATES, this);
if(locationManager!=null){
location=locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
```

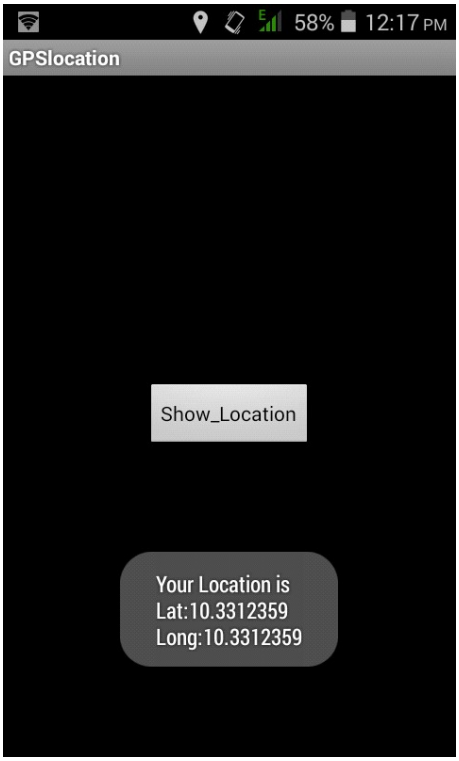
```
if(location!=null){
latitude=location.getLatitude();
longtitude=location.getLongitude();
}
}
}
}
}
catch(Exception e)
{
e.printStackTrace();
}
return location;
}
public void stopUsingGPS(){
if(locationManager!=null){
locationManager.removeUpdates(GPStrace.this);
}
}
public double getLatitude(){
if(location!=null){
latitude=location.getLatitude();
}
return latitude;
}
public double getLongtiude(){
if(location!=null){
longtitude=location.getLatitude();
}
return longtitude;
}
```

```
public boolean canGetLocation(){
return this.canGetLocation;
}
public void showSettingAlert(){
AlertDialog.Builder alertDialog=new AlertDialog.Builder(context);
alertDialog.setTitle("GPS is settings");
alertDialog.setMessage("GPS is not enabled.Do you want to go to setting menu?");
alertDialog.setPositiveButton("settings", new DialogInterface.OnClickListener() {
@Override
public void onClick(DialogInterface dialog,int which){
Intent intent=new Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
context.startActivity(intent);
}
});
alertDialog.setNegativeButton("cancel", new DialogInterface.OnClickListener() {
@Override
public void onClick(DialogInterface dialog, int which) {
// TODO Auto-generated method stub
dialog.cancel();
}
});
alertDialog.show();
}
@Override
public void onLocationChanged(Location location) {
// TODO Auto-generated method stub
}
@Override
public void onProviderDisabled(String provider) {
// TODO Auto-generated method stub
}
```

```
@Override
public void onProviderEnabled(String provider) {
// TODO Auto-generated method stub
}
@Override
public void onStatusChanged(String provider, int status, Bundle extras) {
// TODO Auto-generated method stub
}
@Override
public IBinder onBind(Intent intent) {
// TODO Auto-generated method stub
return null;
}
}
```

11) Now go to main.xml and right click .select run as option and select run configuration.

12) Android output is present in the android emulator as shown in below.



RESULT:

Thus a native application that uses GPS location information has been developed and executed successfully.

EX NO :8 **DATE :**

IMPLEMENT AN APPLICATION THAT WRITES DATA TO THE SD CARD

AIM:

To implement an application that writes data to the SD Card.

PROCEDURE:

1) Open eclipse or android studio and select new android project.
2) Give project name and select next.
3) Choose the android version. Choose the lowest android version (Android 2.2) and select next.
4) Enter the package name. Package name must be two word separated by comma and click finish.
5) Go to package explorer in the left hand side. Select our project.
6) Go to res folder and select layout. Double click the main.xml file. Add the code below

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

7) Now select mainactivity.java file and type the following code.

```
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);
```

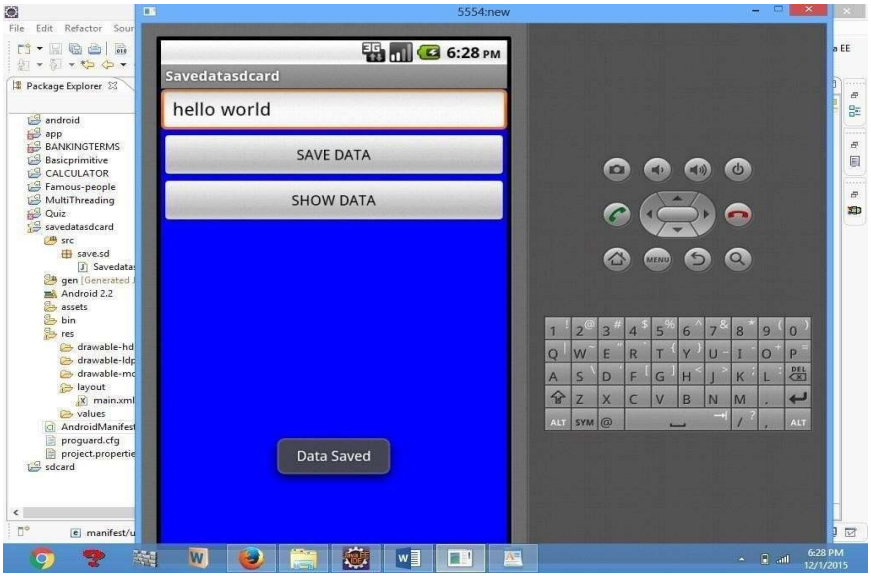
```
final_data+=read_data;
data=new char[100];
}
//display the data in output
Toast.makeText(getBaseContext(),"Message:"+final_data,Toast.LENGTH_LONG).show();
}catch(IOException e){
e.printStackTrace();
}
}catch (FileNotFoundException e){
e.printStackTrace();
}}
});
}}
```

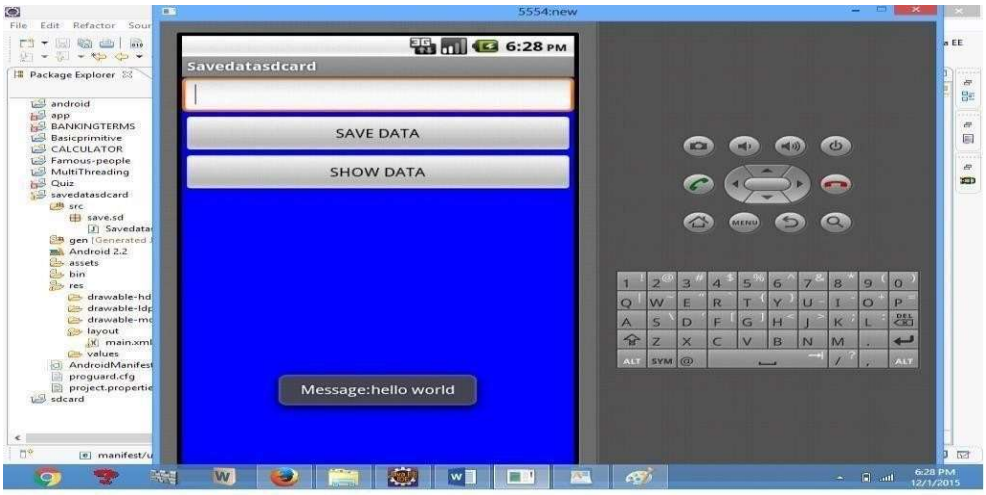
8) Next step is to set permission to write data in sd card. So go to AndroidManifest.xml file.
Copy and paste the following coding. The code should come before <application> tab.

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

9) Now go to main.xml and right click .select run as option and select run configuration.

10) Android output is present in the android emulator as shown in below.





RESULT:

Thus an android application that writes data to the SD Card has been developed and executed successfully.

EX NO :9 **DATE :**

IMPLEMENT AN APPLICATION THAT CREATES AN ALERT UPON RECEIVING A MESSAGE IN ANDROID

AIM:

To implement an application that creates an alert upon receiving a message in android

PROCEDURE:

1) Open eclipse or android studio and select new android project.
2) Give project name and select next.
3) Choose the android version. Choose the lowest android version (Android 2.2) and select next.
4) Enter the package name. Package name must be two word separated by comma and click finish.
5) Go to package explorer in the left hand side. Select our project.
6) Go to res folder and select layout. Double click the main.xml file. Add the code below.

```
<ScrollView xmlns: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
android:layout_height="wrap_content"
android:layout_width="match_parent"
android:gravity="center_horizontal">
<Button
android:id="@+id/submit"
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
android:gravity="center"
android:text="Submit"
/>
</TableRow>
</TableLayout>
</ScrollView>
```

7) Now select mainactivity.java file and type the following code.In my coding maniactivity name is Alert1Activity .

```
package alert1.quiz;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.RadioGroup;
import android.widget.Toast;
import android.view.View.OnClickListener;
import android.widget.RadioGroup.OnCheckedChangeListener;
public class Alert1Activity extends Activity {
private Button btnSubmitQuiz;
int score,ans1,ans2;
/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
```

```
setContentView(R.layout.main);
RadioGroup b1=(RadioGroup)findViewById(R.id.answer1);
b1.setOnCheckedChangeListener(new OnCheckedChangeListener() {
public void onCheckedChanged(RadioGroup group, int checkedId) {
// TODO Auto-generated method stub
switch(checkedId) {
case R.id.answer1A:
ans1 =1;
break;
case R.id.answer1B:
ans1 =2;
break;
case R.id.answer1C:
ans1 =3;
break;
case R.id.answer1D:
ans1=4;
break;
}
}
});
RadioGroup b2=(RadioGroup)findViewById(R.id.answer2);
b2.setOnCheckedChangeListener(new OnCheckedChangeListener() {
public void onCheckedChanged(RadioGroup group, int checkedId) {
// TODO Auto-generated method stub
switch(checkedId) {
case R.id.answer2A:
ans2 =1;
break;
```

```
case R.id.answer2B:
ans2 =2;
break;
case R.id.answer2C:
ans2 =3;
break;
case R.id.answer2D:
ans2=4;
break;
}
}
});
btnSubmitQuiz = (Button) findViewById(R.id.submit);
btnSubmitQuiz.setOnClickListener(new OnClickListener() {
public void onClick(View v) {
AlertDialog.Builder alertDialog = new AlertDialog.Builder(Alert1Activity.this);
// Setting Dialog Title
alertDialog.setTitle("SHOW RESULT");
// Setting Dialog Message
alertDialog.setMessage("Are you sure you want SUBMIT this?");
// Setting Icon to Dialog
// alertDialog.setIcon(R.drawable.tick);
// Setting Positive "Yes" Button
alertDialog.setPositiveButton("YES", new DialogInterface.OnClickListener() {
public void onClick(DialogInterface dialog,int which) {
score =0;
```

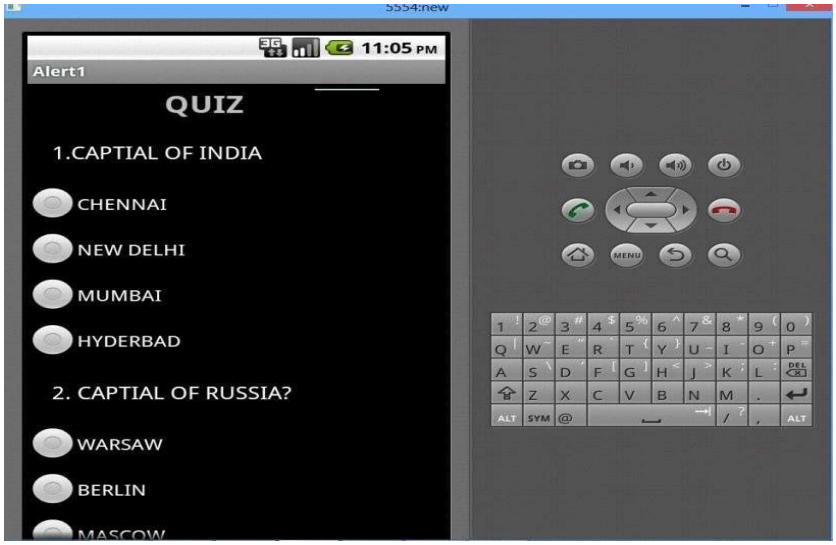
```
// TODO Auto-generated method stub
if(ans1 == 2)
score++;
if(ans2 == 3)
score++;
Toast.makeText(Alert1Activity.this, "Your score is:"+score+" out of 2.",
    Toast.LENGTH_LONG).show();
}
});
// Setting Negative "NO" Button
alertDialog.setNegativeButton("NO", new DialogInterface.OnClickListener() {
public void onClick(DialogInterface dialog, int which) {
// Write your code here to invoke NO event
Toast.makeText(getApplicationContext(), "You clicked NO.CHECK YOUR ANSWER",
    Toast.LENGTH_SHORT).show();
dialog.cancel();
}
});
// Showing Alert Message
alertDialog.show();
}
});
}
}
```

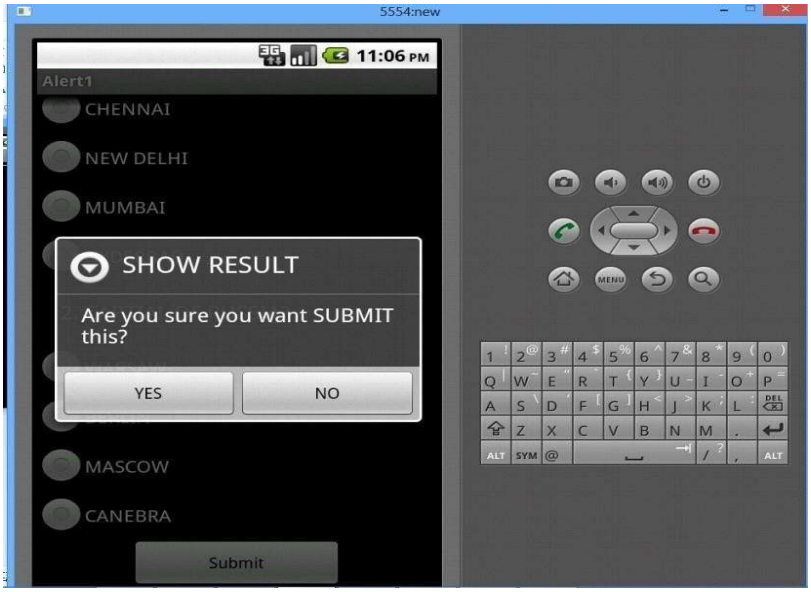
8) Add this in gradle properties

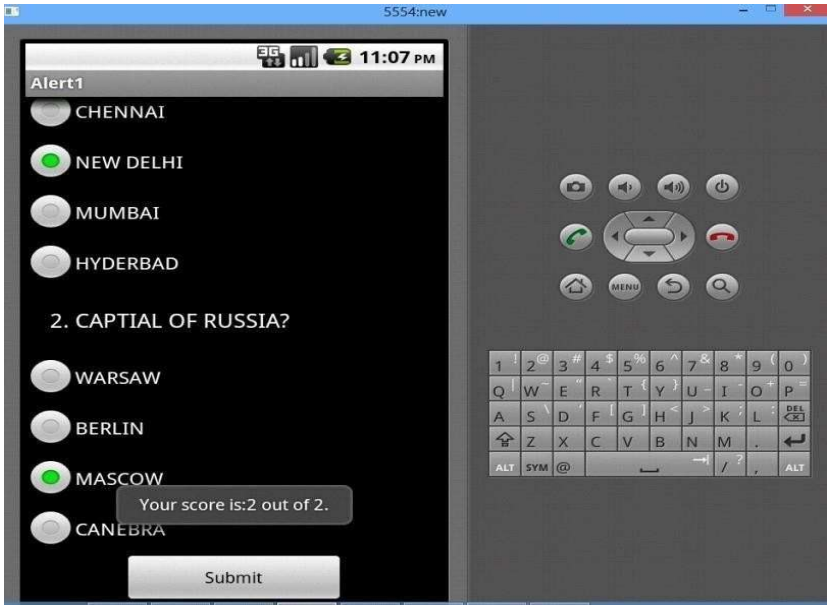
android.nonFinalResIds=false

9) Now go to main.xml and right click .select run as option and select run configuration.

10) Android output is present in the android emulator as shown in below.


5554:new
11:05 PM
Alert1
QUIZ
1.CAPTIAL OF INDIA
CHENNAI
NEW DELHI
MUMBAI
HYDERBAD
2. CAPTIAL OF RUSSIA?
WARSAW
BERLIN
MASCOW


5554:new
11:06 PM
Alert1
CHENNAI
NEW DELHI
MUMBAI
SHOW RESULT
Are you sure you want SUBMIT this?
YES
NO
MASCOW
CANEBRA
Submit



RESULT:

Thus an android application that creates an alert upon receiving a message has been developed and executed successfully.

EX NO:10 **DATE :**

DEVELOP AN APPLICATION THAT MAKES USE OF RSS FEED

AIM:

To develop an application that makes use of RSS Feed.

PROCEDURE:

1) Open eclipse or android studio and select new android project.
2) Give project name and select next.
3) Choose the android version. Choose the lowest android version (Android 2.2) and select next.
4) Enter the package name. Package name must be two word separated by comma and click finish.
5) Go to package explorer in the left hand side. Select our project.
6) Go to res folder and select layout. Double click the main.xml file. Add the code below.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:orientation="vertical" >
<ListView
android:id="@+android:id/list"
android:layout_width="fill_parent"
android:layout_height="wrap_content" >
</ListView>
</LinearLayout>
```

7) Now select mainactivity.java file and type the following code.In my coding maniactivity name is RssFeedActivity.

```
package rss.feed;
//import android.app.Activity;
import java.io.IOException;
import java.io.InputStream;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.ArrayList;
import java.util.List;
import org.xmlpull.v1.XmlPullParser;
import org.xmlpull.v1.XmlPullParserException;
import org.xmlpull.v1.XmlPullParserFactory;
import android.app.ListActivity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.ListView;
public class RssfeedActivity extends ListActivity {
List headlines;
List links;
/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.main);
```

```
headlines = new ArrayList();
links = new ArrayList();
try {
URL url = new URL("http://feeds.pcworld.com/pcworld/latestnews");
XmlPullParserFactory factory = XmlPullParserFactory.newInstance();
factory.setNamespaceAware(false);
XmlPullParser xpp = factory.newPullParser();
// We will get the XML from an input stream
xpp.setInput(getInputStream(url), "UTF_8");
/* We will parse the XML content looking for the "<title>" tag which appears inside the
     "<item>" tag.
* However, we should take in consideration that the rss feed name also is enclosed in a
     "<title>" tag.
* As we know, every feed begins with these lines: "<channel><title>Feed_Name</title> ... "
* so we should skip the "<title>" tag which is a child of "<channel>" tag,
* and take in consideration only "<title>" tag which is a child of "<item>"
*
* In order to achieve this, we will make use of a boolean variable.
*/
boolean insideItem = false;
// Returns the type of current event: START_TAG, END_TAG, etc..
int eventType = xpp.getEventType();
while (eventType != XmlPullParser.END_DOCUMENT) {
if (eventType == XmlPullParser.START_TAG) {
if (xpp.getName().equalsIgnoreCase("item")) {
insideItem = true;
} else if (xpp.getName().equalsIgnoreCase("title")) {
if (insideItem)
```

```
headlines.add(xpp.nextText()); //extract the headline
} else if (xpp.getName().equalsIgnoreCase("link")) {
if (insideItem)
links.add(xpp.nextText()); //extract the link of article
}
}else if(eventType==XmlPullParser.END_TAG &&
     xpp.getName().equalsIgnoreCase("item")){
insideItem=false;
}
eventType = xpp.next(); //move to next element
}
} catch (MalformedURLException e) {
e.printStackTrace();
} catch (XmlPullParserException e) {
e.printStackTrace();
} catch (IOException e) {
e.printStackTrace();
}
// Binding data
ArrayAdapter adapter = new ArrayAdapter(this,
android.R.layout.simple_list_item_1, headlines);
setListAdapter(adapter);
}
public InputStream getInputStream(URL url) {
try {
return url.openConnection().getInputStream();
} catch (IOException e) {
return null;
}
}
```

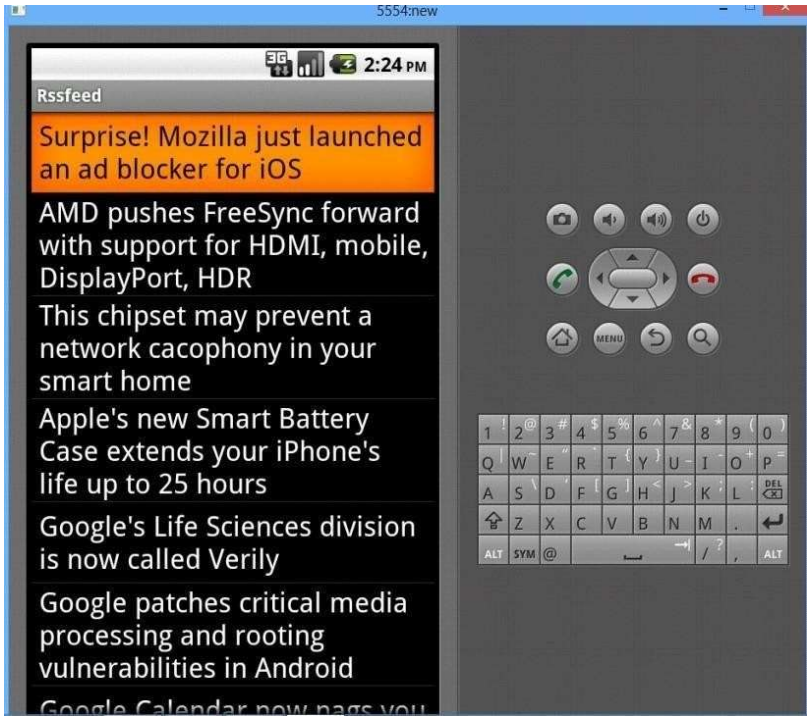
```
@Override
protected void onListItemClick(ListView l, View v, int position, long id) {
Uri uri = Uri.parse((String) links.get(position));
Intent intent = new Intent(Intent.ACTION_VIEW, uri);
startActivity(intent);
}
}
```

8) Go to AndroidManifest.xml file and paste the following code before <appllication> tab

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

9) Now go to main.xml and right click .select run as option and select run configuration.

10) Android output is present in the android emulator as shown in below.



RESULT:

Thus an android application that makes use of RSS Feed has been developed and executed successfully.

EX NO:11 **DATE :**

DEVELOP A MOBILE APPLICATION TO SEND AN EMAIL

AIM:

To Create an Application to send Email using Android Studio.

PROCEDURE:

1) Open eclipse or android studio and select new android project.
2) Give project name and select next.
3) Choose the android version. Choose the lowest android version (Android 2.2) and select next.
4) Enter the package name. Package name must be two word separated by comma and click finish.
5) Go to package explorer in the left hand side. Select our project.
6) Go to res folder and select layout. Double click the main.xml file. Add the code below

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="20dp"
    android:paddingRight="20dp"
    android:orientation="vertical" >
  <EditText
    android:id="@+id/txtTo"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="To"/>
```

```
    <EditText
        android:id="@+id/txtSub"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Subject"/>
    <EditText
        android:id="@+id/txtMsg"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:gravity="top"
        android:hint="Message"/>
    <Button
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:layout_gravity="right"
        android:text="Send"
        android:id="@+id/btnSend"/>
</LinearLayout>
```

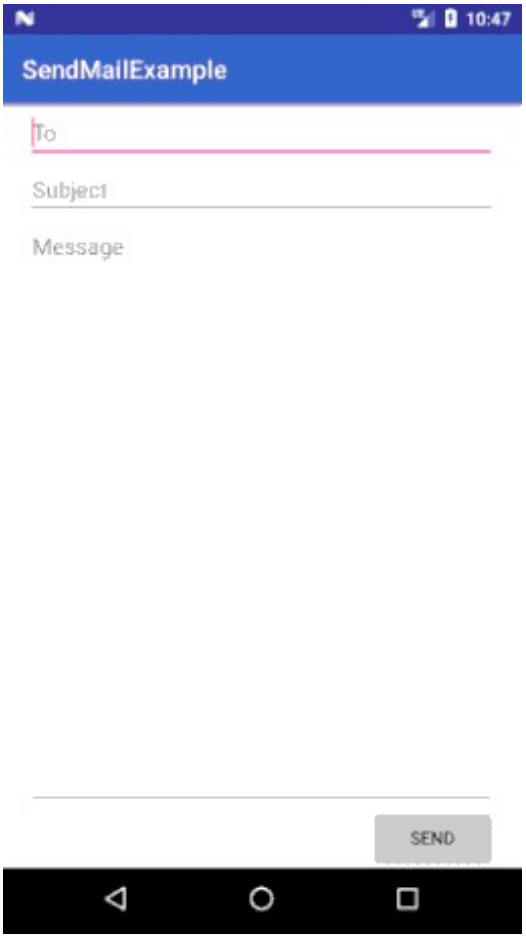
7) Now select mainactivity.java file and type the following code. In my coding main activity.

```
package com.tutlane.sendmailexample;
import android.content.Intent;
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
public class MainActivity extends AppCompatActivity {
    private EditText eTo;
    private EditText eSubject;
```

```
    private EditText eMsg;
    private Button btn;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        eTo = (EditText)findViewById(R.id.txtTo);
        eSubject = (EditText)findViewById(R.id.txtSub);
        eMsg = (EditText)findViewById(R.id.txtMsg);
        btn = (Button)findViewById(R.id.btnSend);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent it = new Intent(Intent.ACTION_SEND);
                it.putExtra(Intent.EXTRA_EMAIL, new String[]{eTo.getText().toString()});
                it.putExtra(Intent.EXTRA_SUBJECT,eSubject.getText().toString());
                it.putExtra(Intent.EXTRA_TEXT,eMsg.getText());
                it.setType("message/rfc822");
                startActivity(Intent.createChooser(it,"Choose Mail App"));
            }
        });
    }
}
```

9) Now go to main.xml and right click .select run as option and select run configuration.

10) Android output is present in the android emulator as shown in below.



RESULT:

Thus an android Application to send Email using Android Studio has been developed and executed successfully