Ex. No: 1 **Android Application that uses GUI components, Font**

Date: **and Colors**

Aim:

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

Procedure:

Step 1: Open Android Studio Application.

Step 2: Click on File -> New -> New project -> enter the Application name as “ex.no.1″ andclick Next and then select the Minimum SDK and click Next -> then select the Empty Activity and click Next and Finish.

Step 3: Go to package explorer in the left hand side. Double click on the project name.

Step 4: Go to res folder and select layout. Double click the activity_main.xml file.

Step 5: Now you can see the Graphical layout window.

Step 6: Drag and drop the following components:

i. Design tab
   a. One TextView with text.
   b. Three Buttons with labeled as Change Font Size, Change Font Color and Change Font Style.(or)

ii. Type the XML commands in the text tab to include the required layout components.

Step 7: Go to package explorer in the left hand side. Select the project name.

Step 8: Go to src folder. Double click the MainActivity.java file.

Step 9: In java file write the activities done by the application such as, actions of buttons.

Step 10: Run the android application.

Program:

Activity_main.xml:

```
<?xml version=”1.0” encoding=”utf-8”?>
<LinearLayout xmlns:android=”http://schemas.android.com/apk/res/android”
   android:orientation=”vertical”
   android:layout_width=”match_parent”
   android:layout_height=”match_parent”>
    <TextView
      android:id=”@+id/textView”
      android:layout_width=”match_parent”
      android:layout_height=”wrap_content”
      android:layout_margin=”30dp”
      android:gravity=”center”
      android:text=”Hello World!”
      android:textSize=”25sp”
      android:textStyle=”bold” />
   <Button
      android:id=”@+id/button1”
      android:layout_width=”match_parent”
      android:layout_height=”wrap_content”
      android:layout_margin=”20dp”
      android:gravity=”center”
      android:text=”Change font size”
      android:textSize=”25sp” />
   <Button
      android:id=”@+id/button2”
      android:layout_width=”match_parent”
      android:layout_height=”wrap_content”
      android:layout_margin=”20dp”
      android:gravity=”center”
```

```
        android:text=”Change color”
        android:textSize=”25sp” />
</LinearLayout>
```

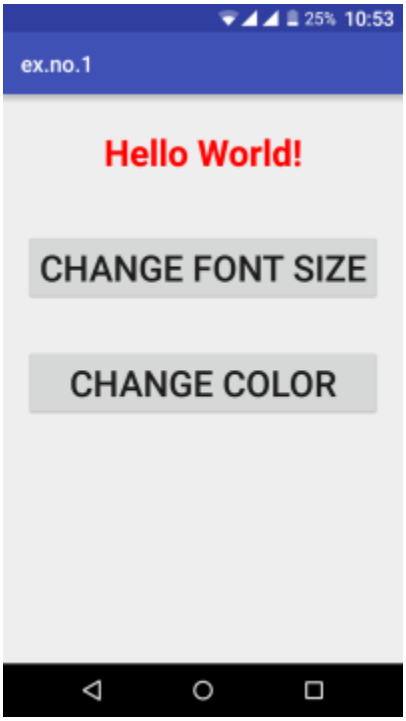
MainActivity.java:

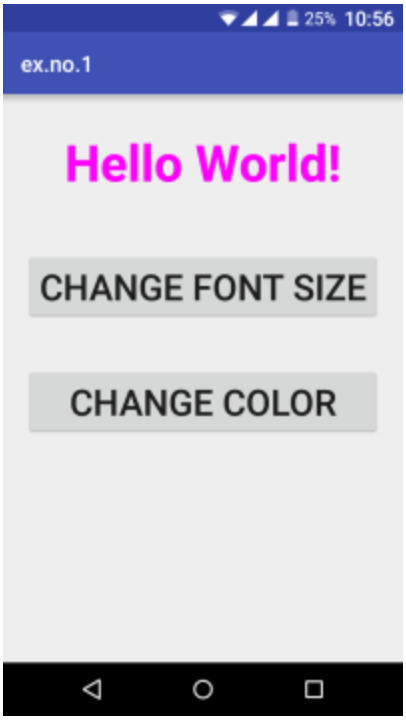
```
package com.example.exno1;
import android.graphics.Color;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
public class MainActivity extends AppCompatActivity
{
   int ch=1;
   float font=30;
   @Override
   protected void onCreate(Bundle savedInstanceState)
   {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      final TextView t= (TextView) findViewById(R.id.textView);
      Button b1= (Button) findViewById(R.id.button1);
      b1.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            t.setTextSize(font);
            font = font + 5;
            if (font == 50)
               font = 30;
         }
```

```
    });
    Button b2= (Button) findViewById(R.id.button2);
    b2.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            switch (ch) {
                case 1:
                    t.setTextColor(Color.RED);
                    break;
                case 2:
                    t.setTextColor(Color.GREEN);
                    break;
                case 3:
                    t.setTextColor(Color.BLUE);
                    break;
                case 4:
                    t.setTextColor(Color.CYAN);
                    break;
                case 5:
                    t.setTextColor(Color.YELLOW);
                    break;
                case 6:
                    t.setTextColor(Color.MAGENTA);
                    break;
            }
            ch++;
            if (ch == 7)
                ch = 1;
        }
    });
}}
```

Output:


25% 10:53
ex.no.1
Hello World!
CHANGE FONT SIZE
CHANGE COLOR


25% 10:53
ex.no.1
Hello World!
CHANGE FONT SIZE
CHANGE COLOR


25% 10:56
ex.no.1
Hello World!
CHANGE FONT SIZE
CHANGE COLOR

Result:

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

Ex. No.:2 **Android Application for Layout Managers and Event**

Date: **Listeners**

Aim:

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

Procedure:

Step 1: Open Android Studio Application.

Step 2: Click on File -> New -> New project -> enter the Application name as "ex.no.2″ and click Next and then select the Minimum SDK and click Next -> then select the Empty Activity and click Next and Finish.

Step 3: Go to package explorer in the left hand side. Double click on the project name.

Step 4: Go to res folder and select layout. Double click the activity_main.xml file.

Step 5: Now you can see the Graphical layout window.

Step 6: Drag and drop the following components:

i. In Design tab include
   a. Three TextViews with texts as Name, Register Number and Department.
   b. Two EditText for the field Name and Register Number.
   c. One Spinner to select the department of candidates.
   d. f. One Button with labeled as SUBMIT(or)

ii. In Text tab, type the XML commands to include the required layout components.

Step 7: Go to package explorer in the left hand side. Select the project name.

Step 8: Go to src folder. Double click the MainActivity.java file.

Step 9: In java file write the activities done by the application such as, actions of buttons.

Step 10: Run the android application.

Program:

Activity_main.xml:

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

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

    <GridLayout
        android:id="@+id/gridLayout"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="100dp"
        android:layout_marginBottom="200dp"
        android:columnCount="2"
        android:rowCount="3">
```

```
<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:layout_row="0"
    android:layout_column="0"
    android:text="Name"
    android:textSize="20sp"
    android:gravity="center"/>

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

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

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

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

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

</GridLayout>
```

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

</RelativeLayout>
```

Activity_second.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.devang.exno2.SecondActivity"
    android:orientation="vertical"
    android:gravity="center">

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

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

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

</LinearLayout>
```

MainActivity.java:

```
package com.example.exno2;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;

public class MainActivity extends AppCompatActivity {
```

```
EditText e1,e2;
Button bt;
Spinner s;

String [] dept_array={"CSE","ECE","IT","Mech","Civil"};

String name,reg,dept;

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

    e1= (EditText) findViewById(R.id.editText);
    e2= (EditText) findViewById(R.id.editText2);

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

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

        ArrayAdapter adapter= new
       ArrayAdapter(MainActivity.this,android.R.layout.simple_spinner_item,dept_arra
       y);
    s.setAdapter(adapter);

    bt.setOnClickListener(new View.OnClickListener() {
       @Override
       public void onClick(View v) {

          name=e1.getText().toString();
```

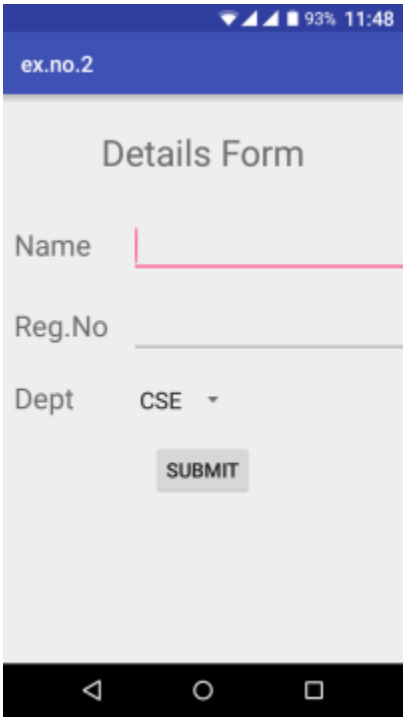
```
            reg=e2.getText().toString();
            dept=s.getSelectedItem().toString();

            Intent i = new Intent(MainActivity.this,SecondActivity.class);

            i.putExtra("name_key", name);
            i.putExtra("reg_key",reg);
            i.putExtra("dept_key", dept);

            startActivity(i);

        }
    });
  }
}
```

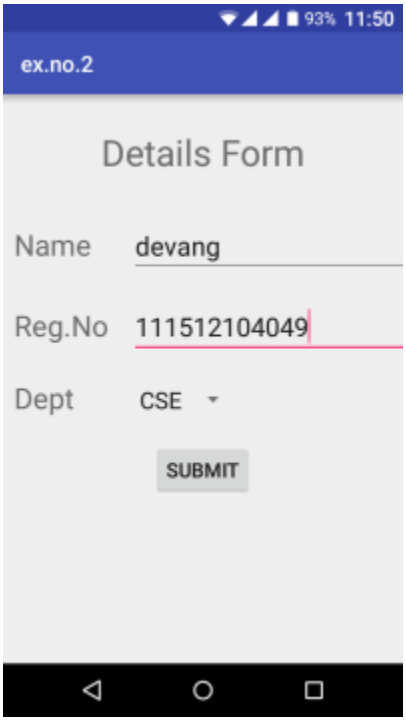
SecondActivity.java:

```
package com.example.exno2;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;

public class SecondActivity extends AppCompatActivity {

  TextView t1,t2,t3;

  String name,reg,dept;
```

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

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



        Intent i = getIntent();

        name=i.getStringExtra("name_key");
        reg=i.getStringExtra("reg_key");
        dept=i.getStringExtra("dept_key");

        t1.setText(name);
        t2.setText(reg);
        t3.setText(dept);

    }
}
```

Output:


93% 11:48
ex.no.2
Details Form
Name
Reg.No
Dept
CSE
SUBMIT


93% 11:50
ex.no.2
Details Form
Name
devang
Reg.No
111512104049
Dept
CSE
SUBMIT


93% 11:50
ex.no.2
devang
111512104049
CSE

Result:

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

Ex. No.:3 Simple Android Application for Native Calculator

Date:

Aim:

To develop a Simple Android Application for Native Calculator.

Procedure:

Step 1: Open Android Studio Application.

Step 2: Click on File -> New -> New project -> enter the Application name as “ex.no.3″ and click Next and then select the Minimum SDK and click Next -> then select the Empty Activity and click Next and Finish.

Step 3: Go to package explorer in the left hand side. Double click on the project name.

Step 4: Go to res folder and select layout. Double click the activity_main.xml file.

Step 5: Now you can see the Graphical layout window.

Step 6: Drag and drop the following components:

i. Design tab
   a. One TextView with to display the result.
   b. Place two EditText for getting two numbers.
   c. Place four buttons (+, -, *, /) to perform calculations. (or)

ii. Type the XML commands in the text tab to include the required layout components.

Step 7: Go to package explorer in the left hand side. Select the project name.

Step 8: Go to src folder. Double click the MainActivity.java file.

Step 9: In java file write the activities done by the application such as, actions of buttons.

Step 10: Run the android application.

Program:

Activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context="com.example.admin.lcalc.MainActivity">
<RelativeLayout
        android:layout_width="368dp"
                android:layout_height="495dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginTop="8dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.0">

<Button
        android:id="@+id/btn_1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_below="@+id/edText1"
        android:layout_marginTop="60dp"
```

```
            android:onClick="PressOne"
            android:text="1"
            android:textSize="18sp"
            tools:ignore="OnClick" />

    <Button
             android:id="@+id/btn_0"
             android:layout_width="wrap_content"
             android:layout_height="wrap_content"
             android:layout_below="@+id/btn_8"
             android:layout_toEndOf="@+id/btn_7"
             android:layout_toRightOf="@+id/btn_7"
             android:text="0"
             android:textSize="18sp" />

    <Button
            android:id="@+id/btn_9"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@+id/btn_6"
            android:layout_toEndOf="@+id/btn_5"
            android:layout_toRightOf="@+id/btn_5"
            android:text="9"
            android:textSize="18sp" />

    <Button
            android:id="@+id/btn_8"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@+id/btn_5"
            android:layout_toEndOf="@+id/btn_7"
```

```
        android:layout_toRightOf="@+id/btn_7"
        android:text="8"
        android:textSize="18sp" />

    <Button
        android:id="@+id/btn_7"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/btn_4"
        android:layout_alignStart="@+id/btn_4"
        android:layout_below="@+id/btn_4"
        android:text="7"
        android:textSize="18sp" />

    <Button
        android:id="@+id/btn_6"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/btn_5"
        android:layout_alignBottom="@+id/btn_5"
        android:layout_toEndOf="@+id/btn_5"
        android:layout_toRightOf="@+id/btn_5"
        android:text="6"
        android:textSize="18sp" />

    <Button
        android:id="@+id/btn_5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/btn_2"
        android:layout_toEndOf="@+id/btn_4"
```

```
        android:layout_toRightOf="@+id/btn_4"
        android:text="5"
        android:textSize="18sp" />

    <Button
        android:id="@+id/btn_4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/btn_1"
        android:layout_alignStart="@+id/btn_1"
        android:layout_below="@+id/btn_1"
        android:text="4"
        android:textSize="18sp" />

    <Button
        android:id="@+id/btn_3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/btn_2"
        android:layout_alignBottom="@+id/btn_2"
        android:layout_toEndOf="@+id/btn_2"
        android:layout_toRightOf="@+id/btn_2"
        android:text="3"
        android:textSize="18sp" />

    <Button
        android:id="@+id/btn_2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/btn_1"
        android:layout_alignBottom="@+id/btn_1"
```

```
        android:layout_toEndOf="@+id/btn_1"
        android:layout_toRightOf="@+id/btn_1"
        android:text="2"
        android:textSize="18sp" />

    <Button
        android:id="@+id/btn_Add"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/btn_6"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:backgroundTint="@android:color/darker_gray"
        android:text="+"
        android:textColor="@android:color/background_light"
        android:textSize="18sp" />

    <Button
        android:id="@+id/btn_Sub"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/btn_Add"
        android:layout_alignStart="@+id/btn_Add"
        android:layout_below="@+id/btn_Add"
        android:backgroundTint="@android:color/darker_gray"
        android:text="-"
        android:textColor="@android:color/background_light"
        android:textSize="18sp" />

    <Button
        android:id="@+id/btn_Mul"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/btn_Sub"
        android:layout_alignStart="@+id/btn_Sub"
        android:layout_below="@+id/btn_6"
        android:backgroundTint="@android:color/darker_gray"
        android:text="*"
        android:textColor="@android:color/background_light"
        android:textSize="18sp" />

    <Button
        android:id="@+id/btn_Div"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/btn_Mul"
        android:layout_alignStart="@+id/btn_Mul"
        android:layout_below="@+id/btn_9"
        android:backgroundTint="@android:color/darker_gray"
        android:text="/"
        android:textColor="@android:color/background_light"
        android:textSize="18sp" />

    <EditText
        android:id="@+id/edText1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
```

```
        android:layout_marginTop="22dp"
        android:ems="10"
        android:inputType="textPersonName"
        android:textAlignment="textEnd"
        android:textSize="24sp"
        android:layout_gravity="end"
        tools:ignore="RtlCompat" />

    <Button
        android:id="@+id/btn_calc"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignTop="@+id/btn_0"
        android:layout_toEndOf="@+id/btn_0"
        android:backgroundTint="@android:color/holo_green_light"
        android:text="="
        android:textColor="@android:color/background_light"
        android:textSize="18sp"
        android:layout_toRightOf="@+id/btn_0"
        tools:ignore="RtlCompat" />

    <Button
        android:id="@+id/btn_dec"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/btn_7"
        android:layout_toLeftOf="@+id/btn_8"
        android:layout_toStartOf="@+id/btn_8"
        android:text="."
        android:textSize="18sp" />
```

```
<Button
        android:id="@+id/btn_clear"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_below="@+id/btn_Div"
        android:backgroundTint="@android:color/holo_blue_dark"
        android:text="clear"
        android:textColor="@android:color/background_light"
        android:textSize="18sp" />
</RelativeLayout>
</android.support.constraint.ConstraintLayout>
```

MainActivity.java

```
package com.example.admin.lcalc;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import com.example.admin.lcalc.R;
public class MainActivity extends AppCompatActivity {
Button btn_1, btn_2, btn_3, btn_4, btn_5, btn_6, btn_7, btn_8, btn_9, btn_0, btn_Add,
btn_Sub,
        btn_Mul, btn_Div, btn_calc, btn_dec, btn_clear;
EditText ed1;
float Value1, Value2;
boolean mAddition, mSubtract, mMultiplication, mDivision ;
@Override
protected void onCreate(Bundle savedInstanceState) {
```

```
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
btn_0 = (Button) findViewById(R.id.btn_0);
btn_1 = (Button) findViewById(R.id.btn_1);
btn_2 = (Button) findViewById(R.id.btn_2);
btn_3 = (Button) findViewById(R.id.btn_3);
btn_4 = (Button) findViewById(R.id.btn_4);
btn_5 = (Button) findViewById(R.id.btn_5);
btn_6 = (Button) findViewById(R.id.btn_6);
btn_7 = (Button) findViewById(R.id.btn_7);
btn_8 = (Button) findViewById(R.id.btn_8);
btn_9 = (Button) findViewById(R.id.btn_9);
btn_Add = (Button) findViewById(R.id.btn_Add);
btn_Div = (Button) findViewById(R.id.btn_Div);
btn_Sub = (Button) findViewById(R.id.btn_Sub);
btn_Mul = (Button) findViewById(R.id.btn_Mul);
btn_calc = (Button) findViewById(R.id.btn_calc);
btn_dec = (Button) findViewById(R.id.btn_dec);
btn_clear = (Button) findViewById(R.id.btn_clear);
ed1 = (EditText) findViewById(R.id.edText1);
btn_0.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
        ed1.setText(ed1.getText()+"0");}});
        btn_1.setOnClickListener(new View.OnClickListener() {
          @Override
        public void onClick(View v) {
                ed1.setText(ed1.getText()+"1");
        }
      });
```

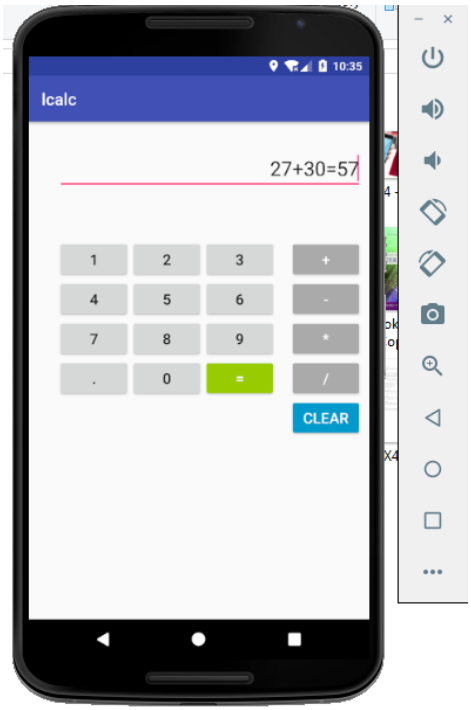
```
btn_2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        ed1.setText(ed1.getText()+"2");
    }
});

btn_3.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        ed1.setText(ed1.getText()+"3");
    }
});

btn_4.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        ed1.setText(ed1.getText()+"4");
    }
});

btn_5.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        ed1.setText(ed1.getText()+"5");
    }
});

btn_6.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
```

```
            ed1.setText(ed1.getText()+"6");
        }
    });
btn_7.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            ed1.setText(ed1.getText()+"7");
        }
    });
    btn_8.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            ed1.setText(ed1.getText()+"8");
        }
    });
    btn_9.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            ed1.setText(ed1.getText()+"9");
        }
    });
    btn_dec.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            ed1.setText(ed1.getText()+".");
        }
    });
    btn_Add.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (ed1 == null){
```

```
            ed1.setText("");
        }else {
            Value1 = Float.parseFloat(ed1.getText() + "");
            mAddition = true;
            ed1.setText(null);
        }
    }
});
btn_Sub.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Value1 = Float.parseFloat(ed1.getText() + "");
        mSubtract = true ;
        ed1.setText(null);
    }
});
btn_Mul.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Value1 = Float.parseFloat(ed1.getText() + "");
        mMultiplication = true ;
        ed1.setText(null);
    }
});
btn_Div.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Value1 = Float.parseFloat(ed1.getText()+"");
        mDivision = true ;
        ed1.setText(null);
    }
```

```
        });
        btn_calc.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Value2 = Float.parseFloat(ed1.getText() + "");
                if (mAddition == true){
                    ed1.setText(Value1 + Value2 +"");
                    mAddition=false;
                }
                if (mSubtract == true){
                    ed1.setText(Value1 - Value2 +"");
                    mSubtract=false;
                }
                if (mMultiplication == true){
                    ed1.setText(Value1 * Value2 + "");
                    mMultiplication=false;
                }
                if (mDivision == true){
                    ed1.setText(Value1 / Value2+"");
                    mDivision=false;
                }
            }
        });
        btn_clear.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                ed1.setText("");
            }
        });
    }
}
```

Output:



Result:

Thus a Simple Android Application to develop a native calculator is developed successfully.

Ex.No.: 4 Android Application to draw Basic Graphical Primitives

Date:

Aim:

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

Procedure:

Step 1: Open Android Studio Application.

Step 2: Click on File -> New -> New project -> enter the Application name as “ex.no.4″ and click Next and then select the Minimum SDK and click Next -> then select the Empty Activity and click Next and Finish.

Step 3: Go to package explorer in the left hand side. Double click on the project name.

Step 4: Go to res folder and select layout. Double click the activity_main.xml file.

Step 5: Now you can see the Graphical layout window.

Step 6: Drag and drop the following components:

i. In Design tab include
    a. One ImageView to view the shapes of graphical primitives on the screen.(or)

ii. In Text tab, type the XML commands to include the required layout components.

Step 7: Go to package explorer in the left hand side. Select the project name.

Step 8: Go to src folder. Double click the MainActivity.java file.

Step 9: In java file write the activities done by the application such as drawing the graphical primitives.

Step 10: Run the android application.

Program:

Activity_main.xml:

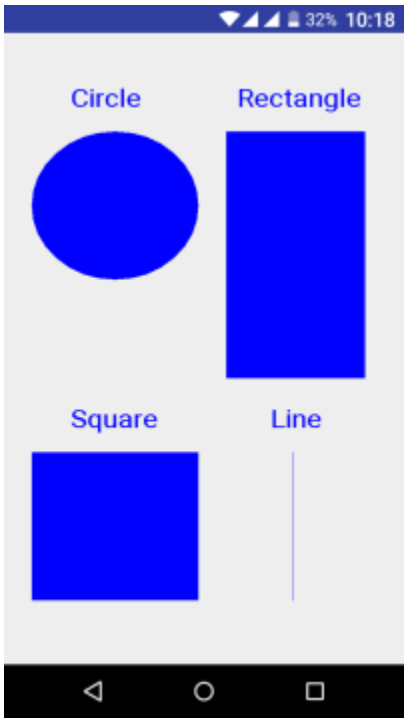
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

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

MainActivity.java:

```
package com.example.exno4;

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

public class MainActivity extends Activity
{
    @Override
    public void onCreate(Bundle savedInstanceState)
```

```
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

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

        ImageView i = (ImageView) findViewById(R.id.imageView);
        i.setBackgroundDrawable(new BitmapDrawable(bg));

        Canvas canvas = new Canvas(bg);

        Paint paint = new Paint();
        paint.setColor(Color.BLUE);
        paint.setTextSize(50);

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

        canvas.drawText("Circle", 120, 150, paint);
        canvas.drawCircle(200, 350, 150, paint);

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

        canvas.drawText("Line", 480, 800, paint);
        canvas.drawLine(520, 850, 520, 1150, paint);
    }
}
```

Output:



Result:

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

Ex. No.: 5 Simple Android Application that makes use of Database

Date:

Aim:

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

Procedure:

Step 1: Open Android Studio Application.

Step 2: Click on File -> New -> New project -> enter the Application name as “ex.no.5″ and click Next and then select the Minimum SDK and click Next -> then select the Empty Activity and click Next and Finish.

Step 3: Go to package explorer in the left hand side. Double click on the project name.

Step 4: Go to res folder and select layout. Double click the activity_main.xml file.

Step 5: Now you can see the Graphical layout window.

Step 6: Drag and drop the following components:

i. In Design tab include
   a. Place the TextView for displaying the Student Details.
   b. Place three TextView for displaying "Enter Roll No","Enter Name","Enter Mark" and create corresponding EditText for RollNo, Name,Marks.
   c. Place five buttons for "INSERT","DELETE","UPDATE","VIEW" and "VIEW ALL".(or)

ii. In Text tab, type the XML commands to include the required layout components.

Step 7: Go to package explorer in the left hand side. Select the project name.

Step 8: Go to src folder. Double click the MainActivity.java file.

Step 9: In java file write the activities done by the application such as actions of buttons.

Step 10: Run the android application.

Program:

Activity_main.xml:

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

   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_x="20dp"
      android:layout_y="110dp"
      android:text="Enter Rollno:"
      android:textSize="20sp" />

   <EditText
      android:id="@+id/Rollno"
      android:layout_width="150dp"
      android:layout_height="wrap_content"
      android:layout_x="175dp"
      android:layout_y="100dp"
      android:inputType="number"
      android:textSize="20sp" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_x="20dp"
    android:layout_y="160dp"
    android:text="Enter Name:"
    android:textSize="20sp" />

<EditText
    android:id="@+id/Name"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="175dp"
    android:layout_y="150dp"
    android:inputType="text"
    android:textSize="20sp" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_x="20dp"
    android:layout_y="210dp"
    android:text="Enter Marks:"
    android:textSize="20sp" />

<EditText
    android:id="@+id/Marks"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="175dp"
```

```
    android:layout_y="200dp"
    android:inputType="number"
    android:textSize="20sp" />

<Button
    android:id="@+id/Insert"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="25dp"
    android:layout_y="300dp"
    android:text="Insert"
    android:textSize="30dp" />

<Button
    android:id="@+id/Delete"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="200dp"
    android:layout_y="300dp"
    android:text="Delete"
    android:textSize="30dp" />

<Button
    android:id="@+id/Update"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="25dp"
    android:layout_y="400dp"
    android:text="Update"
    android:textSize="30dp" />
```

```
    <Button
        android:id="@+id/View"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_x="200dp"
        android:layout_y="400dp"
        android:text="View"
        android:textSize="30dp" />
     <Button
        android:id="@+id/ViewAll"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:layout_x="100dp"
        android:layout_y="500dp"
        android:text="View All"
        android:textSize="30dp" />

</AbsoluteLayout>
```

MainActivity.java:

```
package com.example.exno5;

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

```
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends Activity implements OnClickListener
{
   EditText Rollno,Name,Marks;
   Button Insert,Delete,Update,View,ViewAll;
   SQLiteDatabase db;
   @Override
   public void onCreate(Bundle savedInstanceState)
   {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);

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

      db=openOrCreateDatabase("StudentDB", Context.MODE_PRIVATE, null);
      db.execSQL("CREATE TABLE IF NOT EXISTS student(rollno VARCHAR,name
VARCHAR,marks VARCHAR);");
```

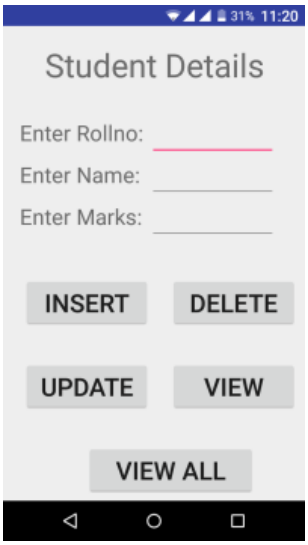
```
    }
    public void onClick(View view)
    {
        if(view==Insert)
        {
            // Checking for empty fields
            if(Rollno.getText().toString().trim().length()==0||
                    Name.getText().toString().trim().length()==0||
                    Marks.getText().toString().trim().length()==0)
            {
                showMessage("Error", "Please enter all values");
                return;
            }
            db.execSQL("INSERT INTO student VALUES('"+Rollno.getText()+"','"+
Name.getText()+"','"+Marks.getText()+"');");
            showMessage("Success", "Record added");
            clearText();
        }
        if(view==Delete)
        {
            if(Rollno.getText().toString().trim().length()==0)
            {
                showMessage("Error", "Please enter Rollno");
                return;
            }
            Cursor c=db.rawQuery("SELECT * FROM student WHERE
 rollno='"+Rollno.getText()+"'", null);
            if(c.moveToFirst())
            {
                db.execSQL("DELETE FROM student WHERE
rollno='"+Rollno.getText()+"'");
```

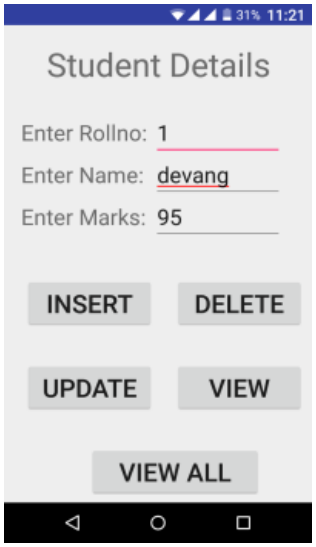
```
            showMessage("Success", "Record Deleted");
          }
          else
          {
            showMessage("Error", "Invalid Rollno");
          }
          clearText();
        }
        if(view==Update)
        {
          if(Rollno.getText().toString().trim().length()==0)
          {
            showMessage("Error", "Please enter Rollno");
            return;
          }
          Cursor c=db.rawQuery("SELECT * FROM student WHERE
rollno='"+Rollno.getText()+"'", null);
          if(c.moveToFirst()) {
            db.execSQL("UPDATE student SET name='" + Name.getText() + "',marks='"
+
Marks.getText() +"' WHERE rollno='"+Rollno.getText()+"'");
            showMessage("Success", "Record Modified");
          }
          else
{
            showMessage("Error", "Invalid Rollno");
          }
          clearText();
        }
        if(view==View)
        {
```

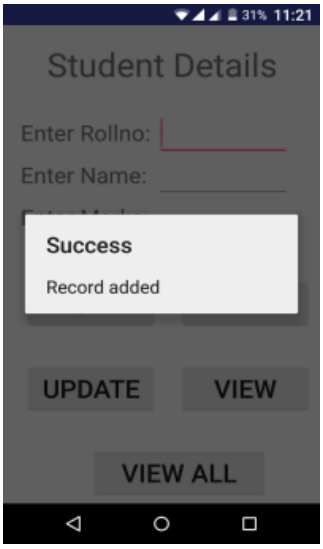
```
            if(Rollno.getText().toString().trim().length()==0)
            {
               showMessage("Error", "Please enter Rollno");
               return;
            }
            Cursor c=db.rawQuery("SELECT * FROM student WHERE rollno='"+
Rollno.getText()+"'", null);
            if(c.moveToFirst())
            {
               Name.setText(c.getString(1));
               Marks.setText(c.getString(2));
            }
            else
            {
               showMessage("Error", "Invalid Rollno");
               clearText();
            }
         }
         if(view==ViewAll)
         {
            Cursor c=db.rawQuery("SELECT * FROM student", null);
            if(c.getCount()==0)
            {
               showMessage("Error", "No records found");
               return;
            }
            StringBuffer buffer=new StringBuffer();
            while(c.moveToNext())
            {
               buffer.append("Rollno: "+c.getString(0)+"\n");
               buffer.append("Name: "+c.getString(1)+"\n");
```

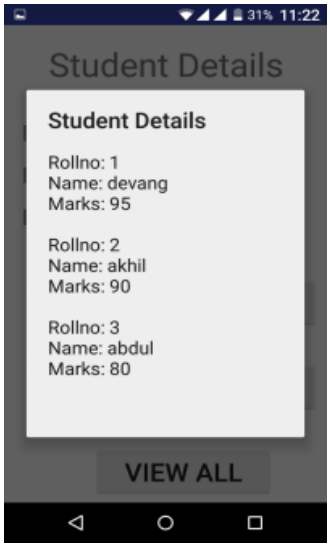
```
                buffer.append("Marks: "+c.getString(2)+"\n\n");
            }
            showMessage("Student Details", buffer.toString());
        }
    }
    public void showMessage(String title,String message)
    {
        Builder builder=new Builder(this);
        builder.setCancelable(true);
        builder.setTitle(title);
        builder.setMessage(message);
        builder.show();
    }
    public void clearText()
    {
        Rollno.setText("");
        Name.setText("");
        Marks.setText("");
        Rollno.requestFocus();
    }
}
```

Output:

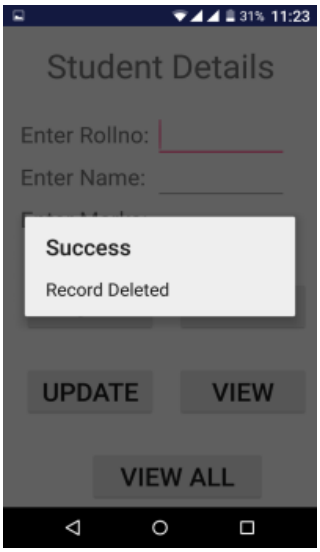

31% 11:20
Student Details
Enter Rollno:
Enter Name:
Enter Marks:
INSERT
DELETE
UPDATE
VIEW
VIEW ALL

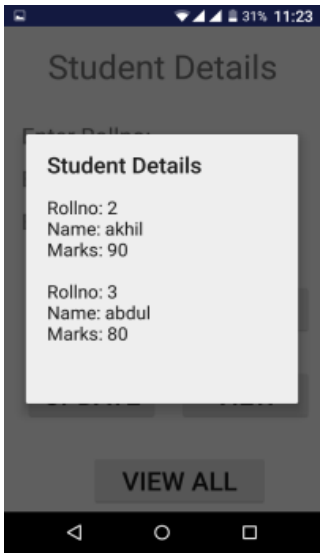

31% 11:21
Student Details
Enter Rollno: 1
Enter Name: devang
Enter Marks: 95
INSERT
DELETE
UPDATE
VIEW
VIEW ALL

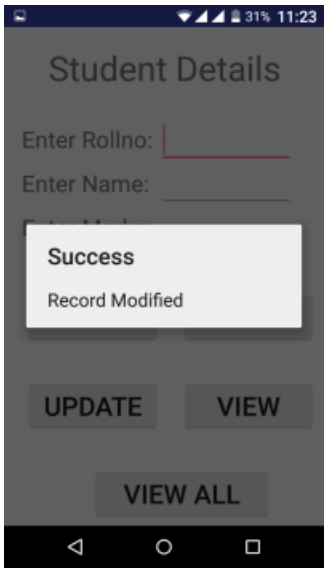

31% 11:21
Student Details
Enter Rollno:
Enter Name:
Success
Record added
UPDATE
VIEW
VIEW ALL


31% 11:22
Student Details
Student Details
Rollno: 1
Name: devang
Marks: 95
Rollno: 2
Name: akhil
Marks: 90
Rollno: 3
Name: abdul
Marks: 80
VIEW ALL

v


31% 11:23
Student Details
Enter Rollno:
Enter Name:
Success
Record Deleted
UPDATE
VIEW
VIEW ALL


31% 11:23
Student Details
Student Details
Rollno: 2
Name: akhil
Marks: 90
Rollno: 3
Name: abdul
Marks: 80
VIEW ALL


31% 11:23
Student Details
Enter Rollno:
Enter Name:
Success
Record Modified
UPDATE
VIEW
VIEW ALL

Result:

Thus a Simple Android Application that makes use of Database is developed successfully.

Ex. No.:6 **Android Application that makes use of RSS Feed**

Date:

Aim:

To develop an Android Application that makes use of RSS Feed.

Procedure:

Step 1: Open Android Studio Application.

Step 2: Click on File -> New -> New project -> enter the Application name as “ex.no.6″ and click Next and then select the Minimum SDK and click Next -> then select the Empty Activity and click Next and Finish.

Step 3: Go to package explorer in the left hand side. Double click on the project name.

Step 4: Go to res folder and select layout. Double click the activity_main.xml file.

Step 5: Now you can see the Graphical layout window.

Step 6: Drag and drop the following components:

i. In Design tab include
    a. Place the ListView.(or)

ii. In Text tab, type the XML commands to include the required layout components.

Step 7: Go to package explorer in the left hand side. Select the project name.

Step 8: Go to src folder. Double click the MainActivity.java file.

Step 9: In java file write the activities done by the application such as different links.

Step 10: Add <uses-permission android:name="android.permission.INTERNET"/> command in AndroidManifest.xml.

Step 11: Run the android application.

Program:

Activity_main.xml:

```
<?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="@+id/listView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
</LinearLayout>
```

AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.exno6" >

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

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

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

```
        </intent-filter>
      </activity>
   </application>

</manifest>
```

MainActivity.java:

```
package com.example.exno6;

import android.app.ListActivity;
import android.content.Intent;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import org.xmlpull.v1.XmlPullParser;
import org.xmlpull.v1.XmlPullParserException;
import org.xmlpull.v1.XmlPullParserFactory;
import java.io.IOException;
import java.io.InputStream;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.ArrayList;
import java.util.List;

public class MainActivity extends ListActivity
{
   List headlines;
```

```
List links;
@Override
protected void onCreate(Bundle savedInstanceState)
{
   super.onCreate(savedInstanceState);
   new MyAsyncTask().execute();
}
class MyAsyncTask extends AsyncTask<Object,Void,ArrayAdapter>
{
   @Override
   protected ArrayAdapter doInBackground(Object[] params)
   {
      headlines = new ArrayList();
      links = new ArrayList();
      try
      {
         URL url = new URL("https://codingconnect.net/feed");
         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");
         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"))
               {
```

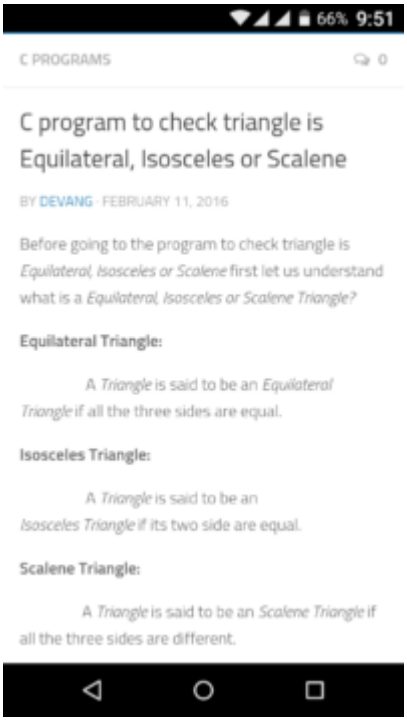
```
                    insideItem = true;
                }
                else if (xpp.getName().equalsIgnoreCase("title"))
                {
                    if (insideItem)
                        headlines.add(xpp.nextText()); //extract the headline
                }
                else if (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();
        }
        return null;
    }
    protected void onPostExecute(ArrayAdapter adapter)
    {
        adapter = new ArrayAdapter(MainActivity.this,
android.R.layout.simple_list_item_1,
headlines);
        setListAdapter(adapter);
    }
  }
  @Override
  protected void onListItemClick(ListView l, View v, int position, long id)
  {
    Uri uri = Uri.parse((links.get(position)).toString());
    Intent intent = new Intent(Intent.ACTION_VIEW, uri);
    startActivity(intent);
  }
  public InputStream getInputStream(URL url)
  {
    try
    {
      return url.openConnection().getInputStream();
    }
    catch (IOException e)
    {
      return null;
    }
  }
}
```

Output:


67% 9:48
Android Application that writes data to the SD Card
Android Application that creates an alert upon receiving a message
Simple Android Application that makes use of Database
Android Application to draw Basic Graphical Primitives
Simple Android Application for Native Calculator
Android Application for Layout Managers and Event Listeners
Android Application that uses GUI components, Font and Colors
C program to print Pascal Triangle without using function
C program to check triangle is Equilateral, Isosceles or Scalene
C program to find Perfect Number or Not using For Loop


67% 9:50
C PROGRAMS
C program to check triangle is Equilateral, Isosceles or Scalene
BY DEVANG · FEBRUARY 11, 2016
Before going to the program to check triangle is Equilateral, Isosceles or Scalene first let us understand what is a Equilateral, Isosceles or Scalene Triangle?
Equilateral Triangle:
A Triangle is said to be an Equilateral


66% 9:51
C PROGRAMS
C program to check triangle is Equilateral, Isosceles or Scalene
BY DEVANG · FEBRUARY 11, 2016
Before going to the program to check triangle is Equilateral, Isosceles or Scalene first let us understand what is a Equilateral, Isosceles or Scalene Triangle?
Equilateral Triangle:
A Triangle is said to be an Equilateral Triangle if all the three sides are equal.
Isosceles Triangle:
A Triangle is said to be an Isosceles Triangle if its two side are equal.
Scalene Triangle:
A Triangle is said to be an Scalene Triangle if all the three sides are different.

Result:

Thus an Android Application that makes use of RSS Feed is developed successfully.

Ex. No.:7 Android Application that implements Multithreading

Date:

Aim:

To develop a Android Application that implements Multithreading.

Procedure:

Step 1: Open Android Studio Application.

Step 2: Click on File -> New -> New project -> enter the Application name as “ex.no.7″ and click Next and then select the Minimum SDK and click Next -> then select the EmptyActivity and click Next and Finish.

Step 3: Go to package explorer in the left hand side. Double click on the project name.

Step 4: Go to res folder and select layout. Double click the activity_main.xml file.

Step 5: Now you can see the Graphical layout window.

Step 6: Drag and drop the following components:

i. In Design tab include
   a. Place the ImageView to view the images on the screen.
   b. Place the two buttons.(or)

ii. In Text tab, type the XML commands to include the required layout components.

Step 7: Go to package explorer in the left hand side. Select the project name.

Step 8: Go to src folder. Double click the MainActivity.java file.

Step 9: In java file write the activities done by the application such as actions of button.

Step 10: Run the android application.

Program:

Activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="250dp"
        android:layout_height="250dp"
        android:layout_margin="50dp"
        android:layout_gravity="center" />
    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_gravity="center"
        android:text="Load Image 1" />
    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_gravity="center"
        android:text="Load image 2" />
</LinearLayout>
```

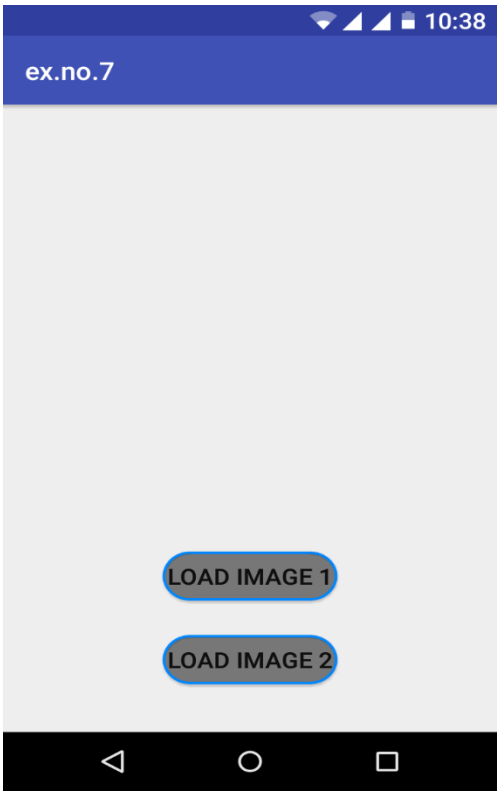
MainActivity.java:

```
package com.example.exno7;
```

```
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
public class MainActivity extends AppCompatActivity
{
   ImageView img;
   Button bt1,bt2;
   @Override
   protected void onCreate(Bundle savedInstanceState)
   {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      bt1 = (Button)findViewById(R.id.button);
      bt2= (Button) findViewById(R.id.button2);
      img = (ImageView)findViewById(R.id.imageView);
      bt1.setOnClickListener(new View.OnClickListener()
      {
         @Override
         public void onClick(View v)
         {
            new Thread(new Runnable()
            {
               @Override
               public void run()
               {
                  img.post(new Runnable()
                  {
                     @Override
                     public void run()
```

```
                {
                    img.setImageResource(R.drawable.india1);
                }
            });
        }
    }).start();
  }
});
bt2.setOnClickListener(new View.OnClickListener()
{
  @Override
  public void onClick(View v)
  {
    new Thread(new Runnable()
    {
      @Override
      public void run()
      {
        img.post(new Runnable()
        {
          @Override
          public void run()
          {
            img.setImageResource(R.drawable.india2);
          }
        });
      }
    }).start();
  }
});
}}
```

Output:


10:38
ex.no.7
LOAD IMAGE 1
LOAD IMAGE 2


10:44
ex.no.7
LOAD IMAGE 1
LOAD IMAGE 2


10:44
ex.no.7
INDIA
LOAD IMAGE 1
LOAD IMAGE 2

Result:

Thus an Android Application that implements Multithreading is developed successfully.

Ex. No.:8 Android Application that uses GPS Location Information

Date:

Aim:

To develop an android application that uses GPS locations information.

Procedure:

Step 1: Open Android Studio Application.

Step 2: Click on File -> New -> New project -> enter the Application name as “ex.no.8″ and click Next and then select the Minimum SDK and click Next -> then select the Empty Activity and click Next and Finish.

Step 3: Go to package explorer in the left hand side. Double click on the project name.

Step 4: Go to res folder and select layout. Double click the activity_main.xml file.

Step 5: Now you can see the Graphical layout window.

Step 6: Drag and drop the following components:

i. In Design tab include
   a. Place one button to get the Current Location.(or)

ii. In Text tab, type the XML commands to include the required layout components.

Step 7: Go to package explorer in the left hand side. Select the project name.

Step 8: Go to src folder. Double click the MainActivity.java file.

Step 9: In java file write the activities done by the application such as finding current location and show them to user using Toast attribute.

Step 10: Get the following permission in AndroidManifest.xml file:

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

Step 11: Run the application.

Program:

Activity_main.xml:

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

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/activity_main"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
tools:context="com.journaldev.gpslocationtracking.MainActivity">

<Button
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/btn"
android:layout_centerInParent="true"
android:text="GET LOCATION" />

</RelativeLayout>
```

AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.journaldev.gpslocationtracking">


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

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

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

MainActivity.java

```
package com.example.exno8;

import android.annotation.TargetApi;

import android.content.DialogInterface;

import android.content.pm.PackageManager;

import android.os.Build;

import android.support.v7.app.AlertDialog;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.util.Log;

import android.view.View;
```

```
import android.widget.Button;
import android.widget.Toast;
import java.util.ArrayList;
import static android.Manifest.permission.ACCESS_COARSE_LOCATION;
import static android.Manifest.permission.ACCESS_FINE_LOCATION;
public class MainActivity extends AppCompatActivity {
    private ArrayList<String> permissionsToRequest;
    private ArrayList<String> permissionsRejected = new ArrayList<>();
    private ArrayList<String> permissions = new ArrayList<>();
    private final static int ALL_PERMISSIONS_RESULT = 101;
    LocationTrack locationTrack;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        permissions.add(ACCESS_FINE_LOCATION);
        permissions.add(ACCESS_COARSE_LOCATION);
        permissionsToRequest = findUnAskedPermissions(permissions);
        //get the permissions we have asked for before but are not granted..
        //we will store this in a global list to access later.
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
            if (permissionsToRequest.size() > 0)
                requestPermissions(permissionsToRequest.toArray(new
```

```
String[permissionsToRequest.size()]), ALL_PERMISSIONS_RESULT);
        }
        Button btn = (Button) findViewById(R.id.btn);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                locationTrack = new LocationTrack(MainActivity.this);
                if (locationTrack.canGetLocation()) {
                    double longitude = locationTrack.getLongitude();
                    double latitude = locationTrack.getLatitude();
                    Toast.makeText(getApplicationContext(), "Longitude:" +
Double.toString(longitude) + "\nLatitude:" + Double.toString(latitude),
Toast.LENGTH_SHORT).show();
                } else {
                    locationTrack.showSettingsAlert();
                }
            }
        });
    }
    private ArrayList<String> findUnAskedPermissions(ArrayList<String> wanted) {
        ArrayList<String> result = new ArrayList<String>();
        for (String perm : wanted) {
            if (!hasPermission(perm)) {
```

```
                result.add(perm);
            }
        }
        return result;
    }
    private boolean hasPermission(String permission) {
        if (canMakeSmores()) {
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
                return (checkSelfPermission(permission) ==
PackageManager.PERMISSION_GRANTED);
            }
        }
        return true;
    }
    private boolean canMakeSmores() {
        return (Build.VERSION.SDK_INT > Build.VERSION_CODES.LOLLIPOP_MR1);
    }
    @TargetApi(Build.VERSION_CODES.M)
    @Override
    public void onRequestPermissionsResult(int requestCode, String[] permissions, int[]
grantResults) {
        switch (requestCode) {
            case ALL_PERMISSIONS_RESULT:
```

```
            for (String perms : permissionsToRequest) {
                if (!hasPermission(perms)) {
                    permissionsRejected.add(perms);
                }
            }
            if (permissionsRejected.size() > 0) {
                if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
                    if (shouldShowRequestPermissionRationale(permissionsRejected.get(0)))
{
                        showMessageOKCancel("These permissions are mandatory for the
application.
Please allow access.",new DialogInterface.OnClickListener() {
@Override
                            public void onClick(DialogInterface dialog, int which) {
                            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
                                    requestPermissions(permissionsRejected.toArray(new
String[permissionsRejected.size()]), ALL_PERMISSIONS_RESULT);


}
}
});
                return;
}
            }
```

```
        }
       break;
    }
}
private void showMessageOKCancel(String message, DialogInterface.OnClickListener
okListener) {
        new AlertDialog.Builder(MainActivity.this)
                .setMessage(message)
                .setPositiveButton("OK", okListener)
                .setNegativeButton("Cancel", null)
                .create()
                .show();
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
        locationTrack.stopListener();
    }
}
```

LocationTrack.java

```
package com.example.exno8;
import android.Manifest;
import android.app.Service;
```

```
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.os.IBinder;
import android.provider.Settings;
import android.support.v4.app.ActivityCompat;
import android.support.v7.app.AlertDialog;
import android.widget.Toast;
public class LocationTrack extends Service implements LocationListener {
    private final Context mContext;
    boolean checkGPS = false;
    boolean checkNetwork = false;
    boolean canGetLocation = false;
    Location loc;
    double latitude;
    double longitude;
    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 LocationTrack(Context mContext) {
        this.mContext = mContext;
        getLocation();
    }
    private Location getLocation() {
        try {

            locationManager = (LocationManager) mContext
.getSystemService(LOCATION_SERVICE);
            // get GPS status
            checkGPS = locationManager
                    .isProviderEnabled(LocationManager.GPS_PROVIDER);
            // get network provider status
            checkNetwork = locationManager
                    .isProviderEnabled(LocationManager.NETWORK_PROVIDER);
            if (!checkGPS && !checkNetwork) {
                Toast.makeText(mContext, "No Service Provider is available",
Toast.LENGTH_SHORT).show();
            } else {
                this.canGetLocation = true;
                // if GPS Enabled get lat/long using GPS Services
                if (checkGPS) {
```

```
            if (ActivityCompat.checkSelfPermission(mContext,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(mContext,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
                // TODO: Consider calling
                //    ActivityCompat#requestPermissions
                // here to request the missing permissions, and then overriding
                //   public void onRequestPermissionsResult(int requestCode, String[]
                //                                          permissions, int[]
grantResults)
                // to handle the case where the user grants the permission. See the
documentation
                // for ActivityCompat#requestPermissions for more details.
            }
            locationManager.requestLocationUpdates(
                LocationManager.GPS_PROVIDER,
                MIN_TIME_BW_UPDATES,
                MIN_DISTANCE_CHANGE_FOR_UPDATES, this);
            if (locationManager != null) {
               loc = locationManager
                     .getLastKnownLocation(LocationManager.GPS_PROVIDER);
               if (loc != null) {
                  latitude = loc.getLatitude();
```

```
                    longitude = loc.getLongitude();
                }
            }
        }
        /*if (checkNetwork) {
            if (ActivityCompat.checkSelfPermission(mContext,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(mContext,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
                // TODO: Consider calling
                //    ActivityCompat#requestPermissions
                // here to request the missing permissions, and then overriding
                //   public void onRequestPermissionsResult(int requestCode, String[]
                //                                          permissions, int[]
grantResults)
                // to handle the case where the user grants the permission. See the
documentation
                // for ActivityCompat#requestPermissions for more details.
            }
            locationManager.requestLocationUpdates(
                    LocationManager.NETWORK_PROVIDER,
                    MIN_TIME_BW_UPDATES,
                    MIN_DISTANCE_CHANGE_FOR_UPDATES, this);
```

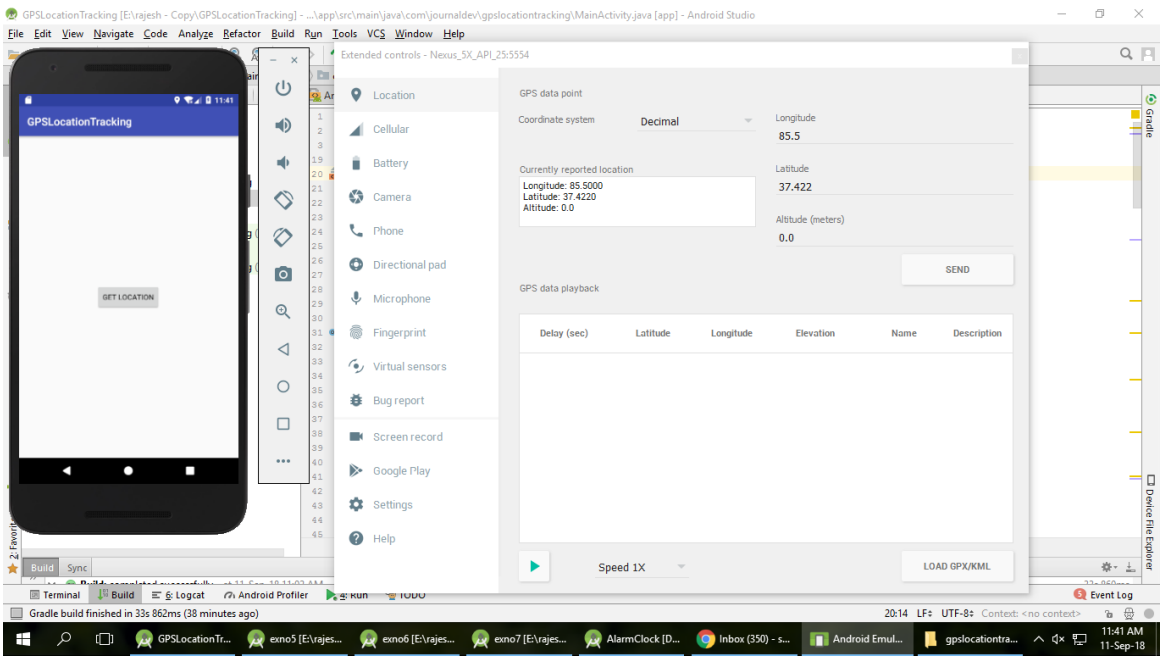
```
                if (locationManager != null) {
                    loc = locationManager

.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
                }

                if (loc != null) {
                    latitude = loc.getLatitude();
                    longitude = loc.getLongitude();
                }
              }*/
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return loc;
    }
    public double getLongitude() {
        if (loc != null) {
            longitude = loc.getLongitude();
        }
        return longitude;
    }
```

```
    public double getLatitude() {
        if (loc != null) {
            latitude = loc.getLatitude();
        }
        return latitude;
    }
    public boolean canGetLocation() {
        return this.canGetLocation;
    }
    public void showSettingsAlert() {
        AlertDialog.Builder alertDialog = new AlertDialog.Builder(mContext);
        alertDialog.setTitle("GPS is not Enabled!");

        alertDialog.setMessage("Do you want to turn on GPS?");
        alertDialog.setPositiveButton("Yes", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
                Intent intent = new
Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
                mContext.startActivity(intent);
            }
        });
        alertDialog.setNegativeButton("No", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
```

```
            dialog.cancel();
        }
    });
    alertDialog.show();
  }
  public void stopListener() {
    if (locationManager != null) {
        if (ActivityCompat.checkSelfPermission(mContext,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(mContext,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
            // TODO: Consider calling
            //    ActivityCompat#requestPermissions
            // here to request the missing permissions, and then overriding
            //   public void onRequestPermissionsResult(int requestCode, String[]
permissions,
            //                                          int[] grantResults)
            // to handle the case where the user grants the permission. See the
documentation
            // for ActivityCompat#requestPermissions for more details.
            return;
        }
        locationManager.removeUpdates(LocationTrack.this);
```

```
        }
    }
  @Override
    public IBinder onBind(Intent intent) {
        return null;
    }
 @Override
    public void onLocationChanged(Location location) {
    }
    @Override
    public void onStatusChanged(String s, int i, Bundle bundle) {
    }
    @Override
    public void onProviderEnabled(String s) {
    }
    @Override
    public void onProviderDisabled(String s) {
    }
}
```

Output:


GPSLocationTracking
GET LOCATION
Extended controls - Nexus_5X_API_25:5554
Location
Cellular
Battery
Camera
Phone
Directional pad
Microphone
Fingerprint
Virtual sensors
Bug report
Screen record
Google Play
Settings
Help
GPS data point
Coordinate system
Decimal
Longitude
85.5
Currently reported location
Longitude: 85.5000
Latitude: 37.4220
Altitude: 0.0
Latitude
37.422
Altitude (meters)
0.0
SEND
GPS data playback
Delay (sec)
Latitude
Longitude
Elevation
Name
Description
Speed 1X
LOAD GPX/KML
Gradle
Device File Explorer
Gradle build finished in 33s 862ms (38 minutes ago)
11:41 AM
11-Sep-18

Result:

Thus an Android Application that makes GPS location information is developed successfully.

Ex. No.: 9 **Android Application that Creates an Alert Upon Receiving a Message**

Date:

Aim:

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

Procedure:

Step 1: Open Android Studio Application.

Step 2: Click on File -> New -> New project -> enter the Application name as “ex.no.9″ and click Next and then select the Minimum SDK and click Next -> then select the Empty Activity and click Next and Finish.

Step 3: Go to package explorer in the left hand side. Double click on the project name.

Step 4: Go to res folder and select layout. Double click the activity_main.xml file.

Step 5: Now you can see the Graphical layout window.

Step 6: This application has no components, because this just generates a notification alone.

Step 7: Go to package explorer in the left hand side. Select the project name.

Step 8: Go to src folder. Double click the MainActivity.java file.

Step 9: In java file write the activities done by the application such as receiving a message and notifying it.

Step 10: Get the following permission in AndroidManifest.xml file:

<uses-permission android:name=”android.permission.RECEIVE_SMS”/>

<uses-permission android:name=”android.permission.READ_SMS”/>

Step 11: Run the application.

Program:

Activity_main.xml:

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

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

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

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

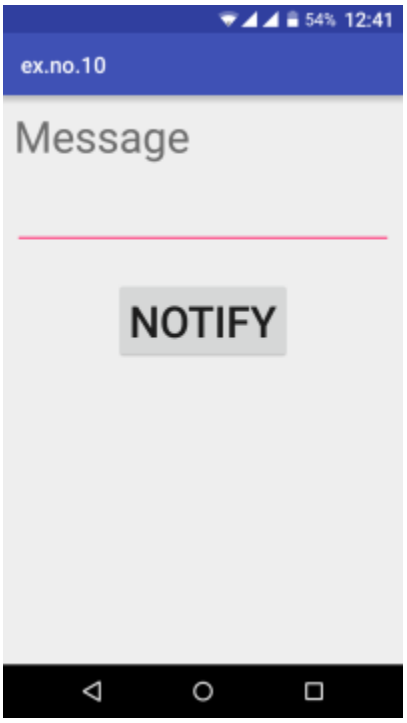
```
</LinearLayout>
```

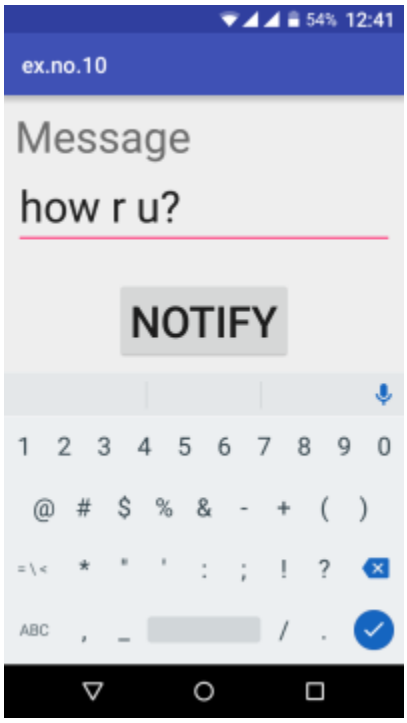
MainActivity.java:

```
package com.example.exno9;

import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

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

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

      notify.setOnClickListener(new View.OnClickListener()
```

```
        {
            @Override
            public void onClick(View v)
            {
                Intent intent = new Intent(MainActivity.this, SecondActivity.class);
                PendingIntent pending = PendingIntent.getActivity(MainActivity.this, 0, intent,
0);
                Notification noti = new
Notification.Builder(MainActivity.this).setContentTitle("New
Message").setContentText(e.getText().toString())
.setSmallIcon(R.mipmap.ic_launcher).setContentIntent(pending).build();
                NotificationManager manager = (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);
                noti.flags |= Notification.FLAG_AUTO_CANCEL;
                manager.notify(0, noti);
            }
        });
    }
}
```

Output:


54% 12:41
ex.no.10
Message
NOTIFY


54% 12:41
ex.no.10
Message
how r u?
NOTIFY


54% 12:41
ex.no.10
Message
how r u?
NOTIFY


12:41 AM
Wednesday, March 16
54%
New Message
12:41 AM
how r u?
how r u?
NOTIFY

Result:

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

Ex. No.:10 **Android Application that creates Alarm Clock**

Date:

Aim:

To develop an Android Application that creates Alarm Clock.

Procedure:

Step 1: Open Android Studio Application.

Step 2: Click on File -> New -> New project -> enter the Application name as “ex.no.10″ and click Next and then select the Minimum SDK and click Next -> then select the Empty Activity and click Next and Finish.

Step 3: Go to package explorer in the left hand side. Double click on the project name.

Step 4: Go to res folder and select layout. Double click the activity_main.xml file.

Step 5: Now you can see the Graphical layout window.

Step 6: Drag and drop the following components:

i. In Design tab include
   a. Place the TimePicker to set the alarm.
   b. Place the ToggleButton to select the A.M or P.M.
   c. Place one button labelled as On or OFF.(or)

ii. In Text tab, type the XML commands to include the required layout components.

Step 7: Go to package explorer in the left hand side. Select the project name.

Step 8: Go to src folder. Double click the MainActivity.java file.

Step 9: In java file write the activities done by the application such as finding current location and show them to user using Toast attribute.

Step 10: Get the following permission in AndroidManifest.xml file:

<uses-permission android:name=”android.permission.WAKE_LOCK”/>

Step 11: Add Alarm class as a receiver in AndroidMainifest.xml file.

Step 12: Run the android application.

Program:

Activity_main.xml:

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

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

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

</LinearLayout>
```

AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   package="com.example.exno11" >

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

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

MainActivity.java:

```
package com.example.exno10;
import android.app.AlarmManager;
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.TimePicker;
import android.widget.Toast;
import android.widget.ToggleButton;
 import java.util.Calendar;

public class MainActivity extends AppCompatActivity
{
   TimePicker alarmTimePicker;
   PendingIntent pendingIntent;
   AlarmManager alarmManager;

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

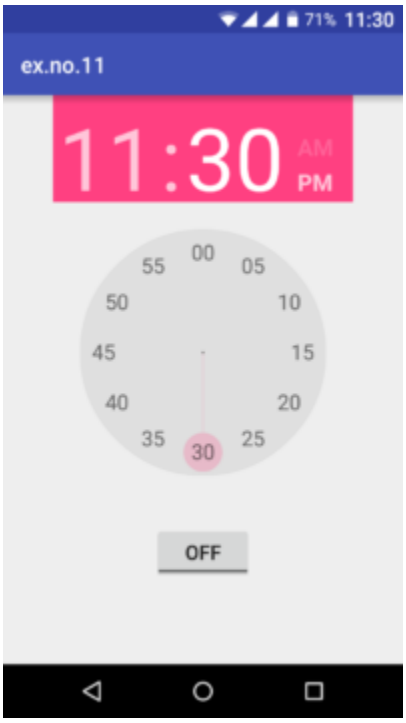
```
        pendingIntent = PendingIntent.getBroadcast(this, 0, intent, 0);

        time=(calendar.getTimeInMillis()-(calendar.getTimeInMillis()%60000));
        if(System.currentTimeMillis()>time)
        {
            if (calendar.AM_PM == 0)
                time = time + (1000*60*60*12);
            else
                time = time + (1000*60*60*24);
        }

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

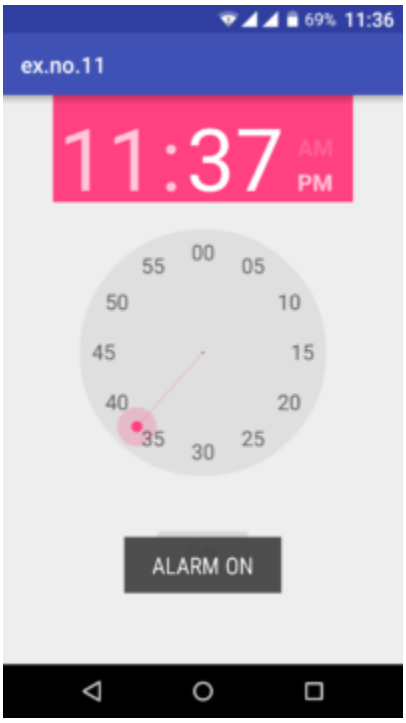
AlarmReceiver.java:

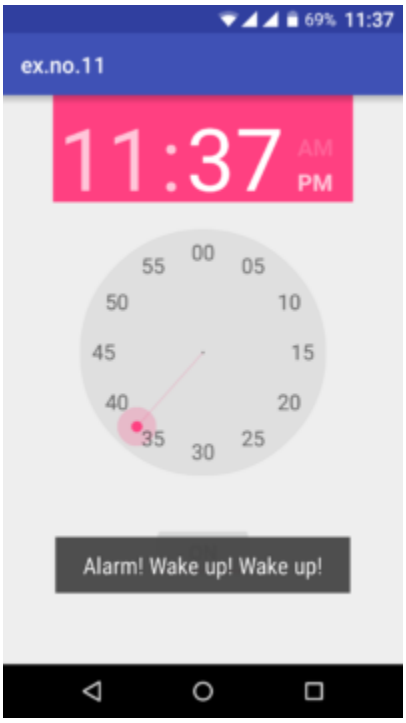
```
package com.example.exno10;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.media.Ringtone;
```

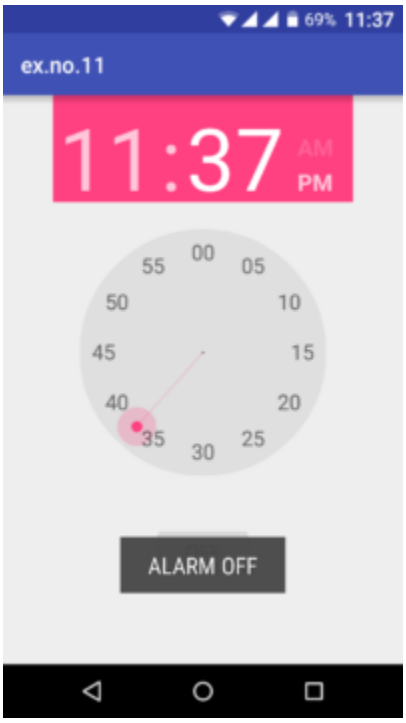
```
import android.media.RingtoneManager;
import android.net.Uri;
import android.widget.Toast;

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

Output:


71% 11:30
ex.no.11
11:30 AM PM
00 05 10 15 20 25 30 35 40 45 50 55
OFF


69% 11:36
ex.no.11
11:37 AM PM
00 05 10 15 20 25 30 35 40 45 50 55
ALARM ON


69% 11:37
ex.no.11
11:37 AM PM
00 05 10 15 20 25 30 35 40 45 50 55
Alarm! Wake up! Wake up!


69% 11:37
ex.no.11
11:37 AM PM
00 05 10 15 20 25 30 35 40 45 50 55
ALARM OFF

Result:

Thus an Android Application that creates Alarm Clock is developed successfully.