Practical No.1

Title: Android program using various UI components

Aim: Create an application to demonstrate various UI components

Introduction

UI Elements

A **View** is an object that draws something on the screen that the user can interact with and a **ViewGroup** is an object that holds other View (and ViewGroup) objects in order to define the layout of the user interface.

You define your layout in an XML file which offers a human-readable structure for the layout, similar to HTML. For example, a simple vertical layout with a text view and a button looks like this –

```
<?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/text"
android:layout_width="wrap_content"
android:layout_height="wrap_content"     android:text="I
am a TextView" />

  <Button android:id="@+id/button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"     android:text="I
am a Button" />
</LinearLayout>
```

Android UI Controls

There are a number of UI controls provided by Android that allow you to build the graphical user interface for your app.

Sr.No.	UI Control & Description
1	TextView This control is used to display text to the user.
2	EditText EditText is a predefined subclass of TextView that includes rich editing capabilities.
3	AutoCompleteTextView The AutoCompleteTextView is a view that is similar to EditText, except that it shows a list of completion suggestions automatically while the user is typing.
4	Button A push-button that can be pressed, or clicked, by the user to perform an action.
5	ImageButton An ImageButton is an AbsoluteLayout which enables you to specify the exact location of its children. This shows a button with an image (instead of text) that can be pressed or clicked by the user.
6	CheckBox An on/off switch that can be toggled by the user. You should use check box when presenting users with a group of selectable options that are not mutually exclusive.
7	ToggleButton An on/off button with a light indicator.

8	RadioButton The RadioButton has two states: either checked or unchecked.
9	RadioGroup A RadioGroup is used to group together one or more RadioButtons.
10	ProgressBar The ProgressBar view provides visual feedback about some ongoing tasks, such as when you are performing a task in the background.
11	Spinner A drop-down list that allows users to select one value from a set.
12	TimePicker The TimePicker view enables users to select a time of the day, in either 24-hour mode or AM/PM mode.
13	DatePicker The DatePicker view enables users to select a date of the day.

Exercise - Create android application and use built in module as well as create own modules

Implementation:

Program:

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:background="@drawable/photo">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="#9309A9"
        android:textStyle="bold"
        app:layout_constraintBottom_toTopOf="@id/button"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/editText" />

    <EditText
        android:id="@+id/editText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="112dp"
        android:hint="Enter a number"
        android:textColor="#9309A9"
        android:inputType="numberDecimal"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />



    <EditText
        android:id="@+id/editText2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:hint="Enter a number"
        android:textColor="#9309A9"
        android:inputType="numberDecimal"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toBottomOf="@id/editText" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="64dp"
        android:onClick="Add"
        android:text="Addition"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.126"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/editText"
        app:layout_constraintVertical_bias="0.214"
        tools:ignore="UsingOnClick" />

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:layout_marginBottom="16dp"
        android:onClick="Sub"
        android:text="Subtraction"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.83"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/editText2"
        app:layout_constraintVertical_bias="0.191"
        tools:ignore="UsingOnClick" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:layout_marginBottom="16dp"
        android:onClick="Multi"
        android:text="Multiplication"

        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.869"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/button1"
        app:layout_constraintVertical_bias="0.201"
        tools:ignore="UsingOnClick" />

    <Button
        android:id="@+id/button4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:layout_marginBottom="16dp"
        android:onClick="Div"
        android:text="Division"

        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.157"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/button"
        app:layout_constraintVertical_bias="0.2"
        tools:ignore="UsingOnClick" />

    <Button
        android:id="@+id/button5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:layout_marginBottom="16dp"
        android:onClick="reset"
        android:text="Reset"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.492"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/button2"
        app:layout_constraintVertical_bias="0.248"
        tools:ignore="UsingOnClick" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

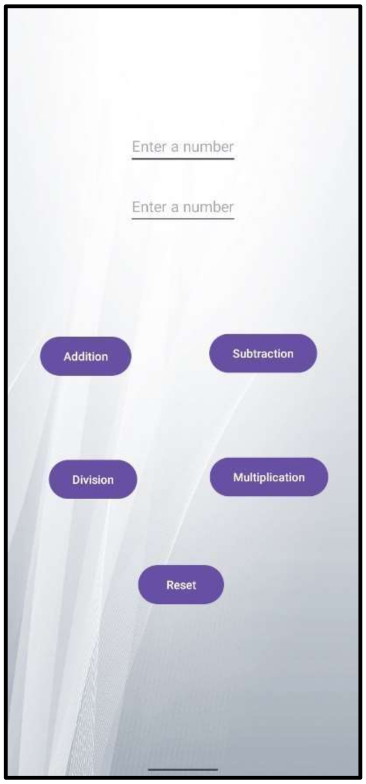
MainActivity.java

```
package com.example.calculator;

import android.animation.ArgbEvaluator;
import android.animation.ObjectAnimator;
import android.animation.ValueAnimator;
import android.os.Bundle;
import android.view.View;
import android.view.animation.Animation;
import android.view.animation.ScaleAnimation;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {
    EditText num1;
    Button add;
    Button sub;
    Button mul;
    Button div;
    EditText num2;
    Button reset;
    TextView textView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        num1 = findViewById(R.id.editText);
        num2 = findViewById(R.id.editText2);
        add = findViewById(R.id.button);
        sub = findViewById(R.id.button1);
        mul = findViewById(R.id.button2);
        div = findViewById(R.id.button4);
        textView = findViewById(R.id.textView);
        reset = findViewById(R.id.button5);

        num1.requestFocus();

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });


    }


    private void animateTextView(TextView textView) {
        // Scale animation
        textView.setScaleX(0);
        textView.setScaleY(0);
        textView.animate()
                .scaleX(1)
                .scaleY(1)
                .setDuration(300)
                .start();

        // Color change animation
        int colorFrom = getResources().getColor(R.color.default_text_color);
        int colorTo = getResources().getColor(R.color.highlight_text_color);
        ValueAnimator colorAnimation = ValueAnimator.ofObject(new ArgbEvaluator(), colorFrom, colorTo);
        colorAnimation.setDuration(300); // duration in milliseconds
        colorAnimation.addUpdateListener(animator -> textView.setTextColor((int) animator.getAnimatedValue()));
        colorAnimation.start();
    }

    public void Add(View v) {
        try {

            String n1 = num1.getText().toString();
            double num1 = Double.parseDouble(n1);
            String n2 = num2.getText().toString();
            double num2 = Double.parseDouble(n2);
            double sum = num1 + num2;
            textView.setText("Addition is: " + sum);
            textView.setVisibility(View.VISIBLE);
            animateTextView(textView);

            Toast.makeText(this, "Addition performed", Toast.LENGTH_SHORT).show();
        } catch (Exception e) {
            Toast.makeText(this, "Error: Enter Values" , Toast.LENGTH_SHORT).show();
        }
    }

    public void Sub(View v) {

        try {
            String n1 = num1.getText().toString();
            double num1 = Double.parseDouble(n1);
            String n2 = num2.getText().toString();
            double num2 = Double.parseDouble(n2);
            double sub = num1 - num2;
            textView.setText("Subtraction is: " + sub);
            textView.setVisibility(View.VISIBLE);
            animateTextView(textView);
            Toast.makeText(this, "Subtraction performed", Toast.LENGTH_SHORT).show();
        } catch (Exception e) {
            Toast.makeText(this, "Error: Enter Values", Toast.LENGTH_SHORT).show();
        }
    }

    public void Multi(View v) {

        try {
            String n1 = num1.getText().toString();
            double num1 = Double.parseDouble(n1);
            String n2 = num2.getText().toString();
            double num2 = Double.parseDouble(n2);
            double multi = num1 * num2;
            textView.setText("Multiplication is: " + multi);
            textView.setVisibility(View.VISIBLE);
            animateTextView(textView);
            Toast.makeText(this, "Multiplication performed", Toast.LENGTH_SHORT).show();
```

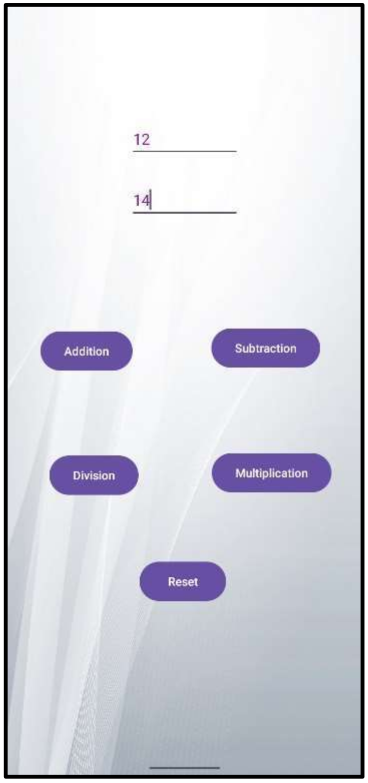
```
        } catch (Exception e) {
            Toast.makeText(this, "Error: Enter Values" , Toast.LENGTH_SHORT).show();
        }
    }

    public void Div(View v) {

        try {
            String n1 = num1.getText().toString();
            double num1 = Double.parseDouble(n1);
            String n2 = num2.getText().toString();
            double num2 = Double.parseDouble(n2);
            double div = num1 / num2;
            textView.setText("Division is: " + div);
            textView.setVisibility(View.VISIBLE);
            animateTextView(textView);
            Toast.makeText(this, "Division performed", Toast.LENGTH_SHORT).show();
        } catch (Exception e) {
            Toast.makeText(this, "Error: Enter Values", Toast.LENGTH_SHORT).show();
        }
    }

    public void reset(View v) {

        try {
            num1.setText("");
            num2.setText("");
            textView.setText("");
            num1.requestFocus();
            Toast.makeText(this, "Reset performed", Toast.LENGTH_SHORT).show();
        } catch (Exception e) {
            Toast.makeText(this, "Error: Cannot Reset", Toast.LENGTH_SHORT).show();
        }
    }


}
```

Output:

UI Design:

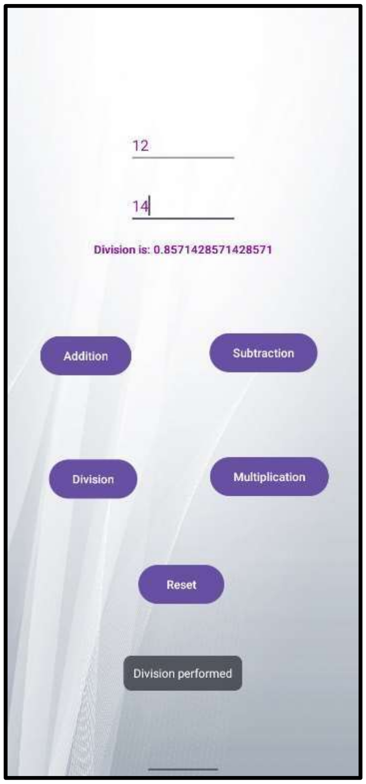


Actual Output:



12

14

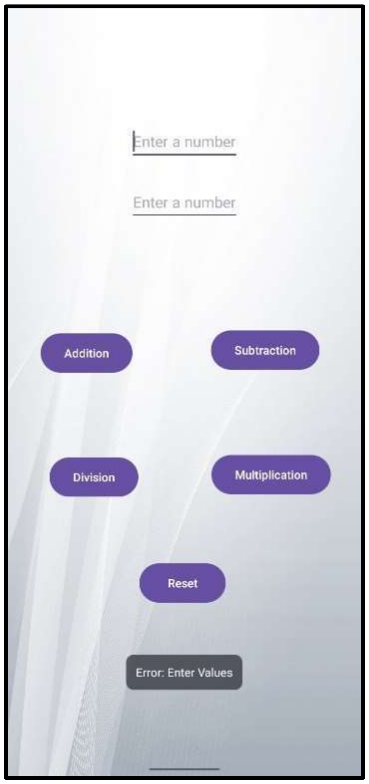


12

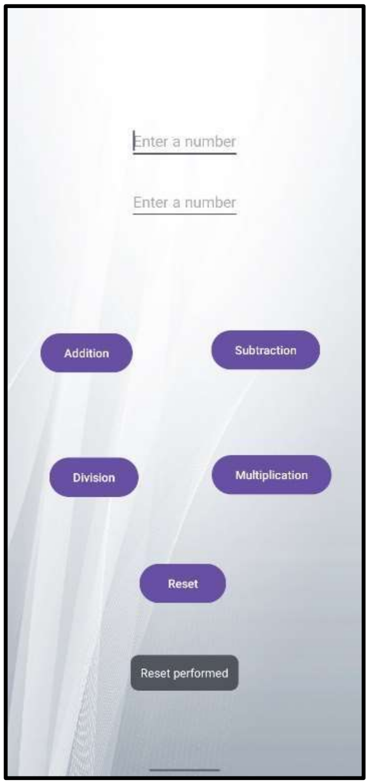
14

Division is: 0.8571428571428571

Division performed



Error: Enter Values



Reset performed

Conclusion -

This practical demonstrated the creation of an Android application using various UI components like TextView, EditText, and Button. We built a basic calculator app with arithmetic and added animations for a better user experience, highlighting key concepts in Android UI design and development.