Ex. No. : 02 **Date:**

Register No.: 221701029 **Name: Keerthana V**

Calculator

Aim

Develop a scientific calculator to perform arithmetic and mathematical functions using Math class.

Procedure:

1. Create a new Android project named exp2 in Android Studio.
2. Design the layout using Buttons and a TextView in activity_main.xml.
3. Add buttons for digits 0–9, operators (+, -, *, /, %, .) and functions (sin, cos, tan, sqrt, log, pow).
4. Set onClick listeners for all the buttons in MainActivity.java.
5. When a number or operator button is clicked, update the input string.
6. When a scientific function button is clicked, apply the corresponding Math class function.
7. When the equal button is clicked, evaluate the full expression.
8. Display the result in the TextView.
9. Handle invalid inputs and show error messages using Toast.

AndroidManifest.xml

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

  <application
    android:allowBackup="true"
    android:dataExtractionRules="@xml/data_extraction_rules"
    android:fullBackupContent="@xml/backup_rules"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/Theme.Exp2"
    tools:targetApi="31">
    <activity
      android:name=".MainActivity"
      android:exported="true">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

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">

  <!-- Display Screen -->
  <TextView
    android:id="@+id/display"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="0"
    android:textSize="30sp"
    android:textColor="#000000"
    android:gravity="end"
    android:padding="20dp"/>

  <!-- Calculator Buttons -->
  <GridLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/display"
    android:columnCount="4"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="30dp">

    <!-- Row 1 -->
    <Button
      android:id="@+id/button_7"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="7"/>
    <Button
      android:id="@+id/button_8"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="8"/>
    <Button
      android:id="@+id/button_9"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="9"/>
    <Button
      android:id="@+id/button_divide"
```


```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="/"/>

<!-- Row 2 -->
<Button
    android:id="@+id/button_4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="4"/>
<Button
    android:id="@+id/button_5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="5"/>
<Button
    android:id="@+id/button_6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="6"/>
<Button
    android:id="@+id/button_multiply"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="*"/>

<!-- Row 3 -->
<Button
    android:id="@+id/button_1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="1"/>
<Button
    android:id="@+id/button_2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="2"/>
<Button
    android:id="@+id/button_3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="3"/>
<Button
```

```
    android:id="@+id/button_minus"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="-"/>

<!-- Row 4 -->
<Button
    android:id="@+id/button_0"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="0"/>
<Button
    android:id="@+id/button_dot"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="."/>
<Button
    android:id="@+id/button_equal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="="/>
<Button
    android:id="@+id/button_plus"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="+"/>

<!-- Row 5: Scientific Functions -->
<Button
    android:id="@+id/button_sin"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="sin"/>
<Button
    android:id="@+id/button_cos"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="cos"/>
<Button
    android:id="@+id/button_tan"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="tan"/>
<Button
```


```
        android:id="@+id/button_sqrt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="√"/>

    <!-- Row 6: More Functions -->
    <Button
        android:id="@+id/button_log"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="log"/>
    <Button
        android:id="@+id/button_pow"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="^"/>
    <Button
        android:id="@+id/button_mod"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="%"/>
    <Button
        android:id="@+id/button_clear"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="C"/>
  </GridLayout>

</RelativeLayout>
```

MainActivity.kt

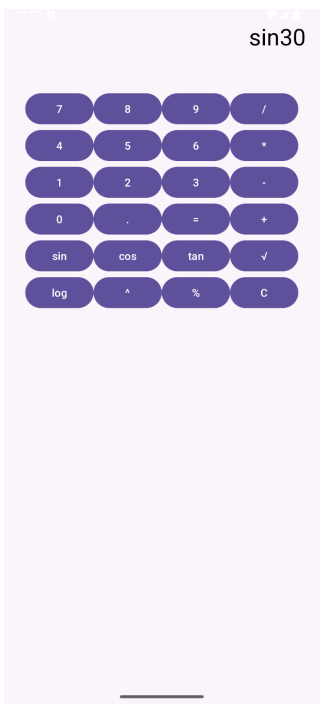
```
package com.example.exp2;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

   private TextView display;
   private String currentInput = "";
   private double result = 0;

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

      display = findViewById(R.id.display);

      // Number buttons
      setButtonClickListener(R.id.button_1, "1");
      setButtonClickListener(R.id.button_2, "2");
      setButtonClickListener(R.id.button_3, "3");
      setButtonClickListener(R.id.button_4, "4");
      setButtonClickListener(R.id.button_5, "5");
      setButtonClickListener(R.id.button_6, "6");
      setButtonClickListener(R.id.button_7, "7");
      setButtonClickListener(R.id.button_8, "8");
      setButtonClickListener(R.id.button_9, "9");
      setButtonClickListener(R.id.button_0, "0");
      setButtonClickListener(R.id.button_dot, ".");

      // Operator buttons
      setButtonClickListener(R.id.button_plus, "+");
      setButtonClickListener(R.id.button_minus, "-");
      setButtonClickListener(R.id.button_multiply, "*");
      setButtonClickListener(R.id.button_divide, "/");

      // Scientific function buttons
      setButtonClickListener(R.id.button_sin, "sin");
```

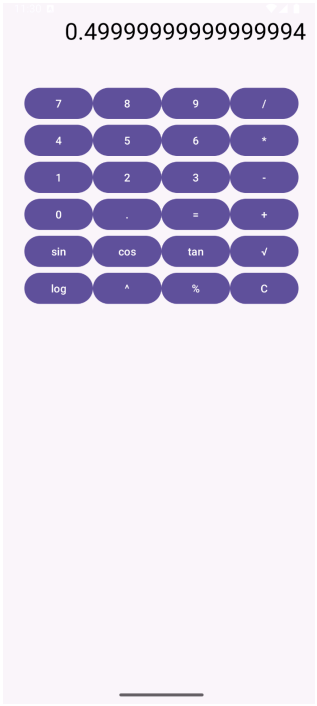

```
        setButtonClickListener(R.id.button_cos, "cos");
        setButtonClickListener(R.id.button_tan, "tan");
        setButtonClickListener(R.id.button_sqrt, "sqrt");
        setButtonClickListener(R.id.button_log, "log");
        setButtonClickListener(R.id.button_pow, "pow");
        setButtonClickListener(R.id.button_mod, "%");

        // Equals button to evaluate expression
        Button buttonEqual = findViewById(R.id.button_equal);
        buttonEqual.setOnClickListener(v -> evaluateExpression());

        // Clear button
        Button buttonClear = findViewById(R.id.button_clear);
        buttonClear.setOnClickListener(v -> clearInput());
    }

    // Set the click listener for number and operator buttons
    private void setButtonClickListener(int buttonId, String value) {
        Button button = findViewById(buttonId);
        button.setOnClickListener(v -> appendToInput(value));
    }

    // Append text to the display input
    private void appendToInput(String value) {
        currentInput += value;
        display.setText(currentInput);
    }

    // Evaluate the mathematical expression when "=" is pressed
    private void evaluateExpression() {
        try {
                        if (currentInput.contains("sin") || currentInput.contains("cos") ||
currentInput.contains("tan") ||
                                currentInput.contains("sqrt") || currentInput.contains("log") ||
currentInput.contains("pow")) {

                result = handleScientificFunctions(currentInput);
            } else {
                result = eval(currentInput); // Evaluate basic arithmetic expression
            }
            display.setText(String.valueOf(result));
        } catch (Exception e) {
            display.setText("Error");
```

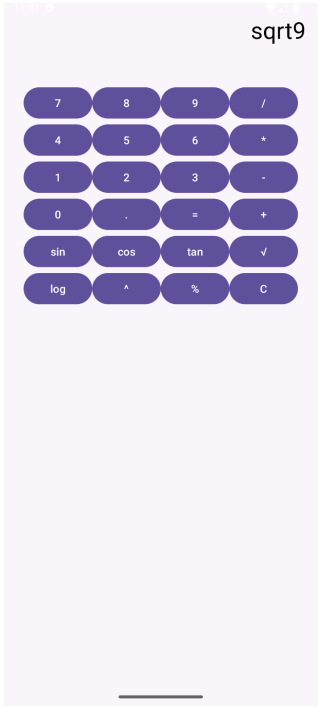
```
            Toast.makeText(this, "Invalid Input", Toast.LENGTH_SHORT).show();
        }
    }

    // Handle scientific functions like sin, cos, tan, etc.
    private double handleScientificFunctions(String input) {
        double val = 0;
        if (input.contains("sin")) {
            val = Math.sin(Math.toRadians(Double.parseDouble(input.substring(3))));
        } else if (input.contains("cos")) {
            val = Math.cos(Math.toRadians(Double.parseDouble(input.substring(3))));
        } else if (input.contains("tan")) {
            val = Math.tan(Math.toRadians(Double.parseDouble(input.substring(3))));
        } else if (input.contains("sqrt")) {
            val = Math.sqrt(Double.parseDouble(input.substring(4)));
        } else if (input.contains("log")) {
            val = Math.log(Double.parseDouble(input.substring(3)));
        } else if (input.contains("pow")) {
            // Example: "pow(2,3)" means 2 raised to the power of 3
            String[] parts = input.substring(4, input.length() - 1).split(",");
                                        val = Math.pow(Double.parseDouble(parts[0]),
Double.parseDouble(parts[1]));
        }
        return val;
    }

    // Simple method for arithmetic evaluation (for +, -, *, /)
    private double eval(String expression) {
        String[] tokens = expression.split(" ");
        double num1 = Double.parseDouble(tokens[0]);
        String operator = tokens[1];
        double num2 = Double.parseDouble(tokens[2]);

        switch (operator) {
            case "+":
                return num1 + num2;
            case "-":
                return num1 - num2;
            case "*":
                return num1 * num2;
            case "/":
                return num1 / num2;
            case "%":
                return num1 % num2;
```

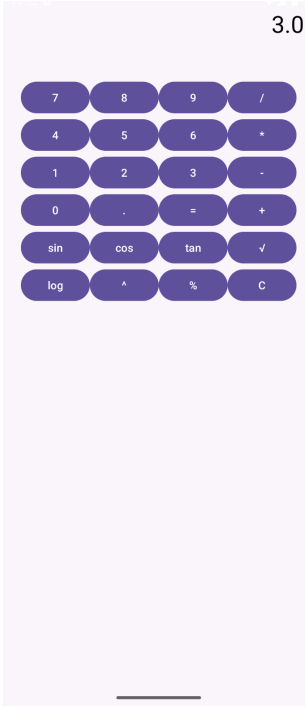

```
            default:
                return 0;
        }
    }

    // Clear the input
    private void clearInput() {
        currentInput = "";
        display.setText("0");
    }
}
```

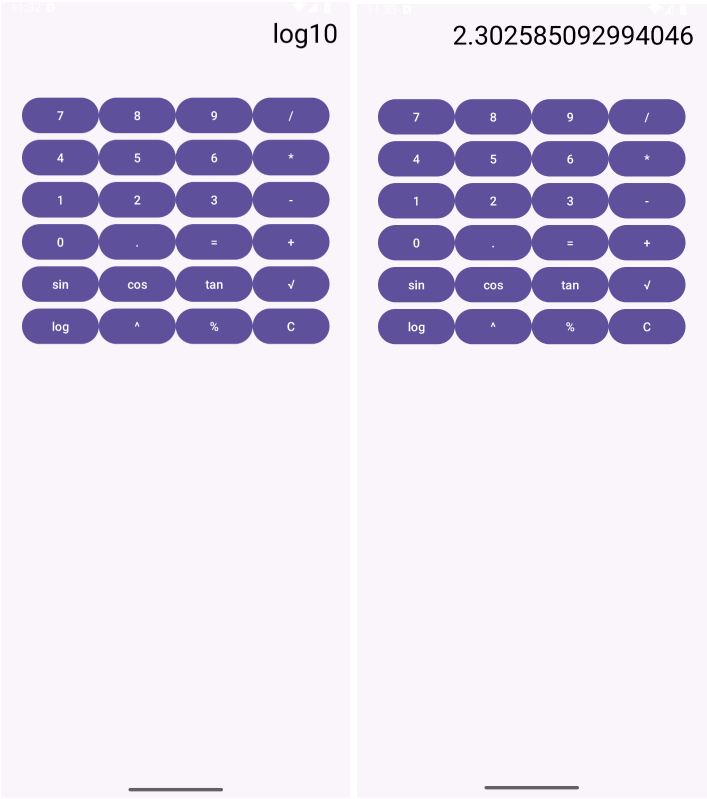
Output


sin30
7 8 9 /
4 5 6 *
1 2 3 -
0 . = +
sin cos tan √
log ^ % C


0.49999999999999994
7 8 9 /
4 5 6 *
1 2 3 -
0 . = +
sin cos tan √
log ^ % C


sqrt9
7 8 9 /
4 5 6 *
1 2 3 -
0 . = +
sin cos tan √
log ^ % C


3.0
7 8 9 /
4 5 6 *
1 2 3 -
0 . = +
sin cos tan √
log ^ % C



Result:

The application successfully performs arithmetic and scientific calculations using the Math class and displays the result.