Ex. No. : 02 **Date: 28/02/2025**

Register No.: 221701023 **Name: JANAKIRAMAN K**

Calculator

Aim

Develop a scientific calculator to perform arithmetic and mathematical functions using Math class. [Your scientific calculator should contain +, *, /, =, cos, sin, tan, pow, sqrt, log, tan and mod].

Procedure:

Step 1 : *File -> NewProject*

Provide the application name and Click "Next"

Step 2 : *Select the target android devices*

Select the minimum SDK to run the application. Click "Next".

Step 3 : *Choose the activity for the application (By default choose "Blank Activity).*

Click "Next".

Step 4 : *Enter activity name and click.*

Step 5 : *Edit the program.*

Step 6 : *Run the application, 2-ways to run the application.*

1. Running through emulator

2. Running through mobile device

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._2exp23"
        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"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:orientation="vertical"
        android:padding="16dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <TextView
            android:id="@+id/tvInput"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textSize="32sp"
            android:padding="16dp"
            android:gravity="end"
            android:background="#EEEEEE"
            android:text=""/>

        <GridLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:columnCount="4"
            android:rowCount="6"
            android:layout_marginTop="16dp"
            android:alignmentMode="alignMargins"
            android:useDefaultMargins="true">

            <!-- Row 1 -->
            <Button android:text="7" android:onClick="onDigitClick"/>
            <Button android:text="8" android:onClick="onDigitClick"/>
            <Button android:text="9" android:onClick="onDigitClick"/>
            <Button android:text="/" android:onClick="onOperatorClick"/>

            <!-- Row 2 -->
            <Button android:text="4" android:onClick="onDigitClick"/>
            <Button android:text="5" android:onClick="onDigitClick"/>
            <Button android:text="6" android:onClick="onDigitClick"/>
            <Button android:text="*" android:onClick="onOperatorClick"/>

            <!-- Row 3 -->
            <Button android:text="1" android:onClick="onDigitClick"/>
```

```
        <Button android:text="2" android:onClick="onDigitClick"/>
        <Button android:text="3" android:onClick="onDigitClick"/>
        <Button android:text="-" android:onClick="onOperatorClick"/>

        <!-- Row 4 -->
        <Button android:text="0" android:onClick="onDigitClick"/>
        <Button android:text="." android:onClick="onDigitClick"/>
        <Button android:text="=" android:onClick="onEqualClick"/>
        <Button android:text="+" android:onClick="onOperatorClick"/>

        <!-- Row 5 -->
        <Button android:text="C" android:onClick="onClearClick"/>
        <Button android:text="mod" android:onClick="onOperatorClick"/>
        <Button android:text="pow" android:onClick="onOperatorClick"/>
        <Button android:text="sqrt" android:onClick="onFunctionClick"/>

        <!-- Row 6 -->
        <Button android:text="log" android:onClick="onFunctionClick"/>
        <Button android:text="sin" android:onClick="onFunctionClick"/>
        <Button android:text="cos" android:onClick="onFunctionClick"/>
        <Button android:text="tan" android:onClick="onFunctionClick"/>

    </GridLayout>
  </LinearLayout>
</ScrollView>
```

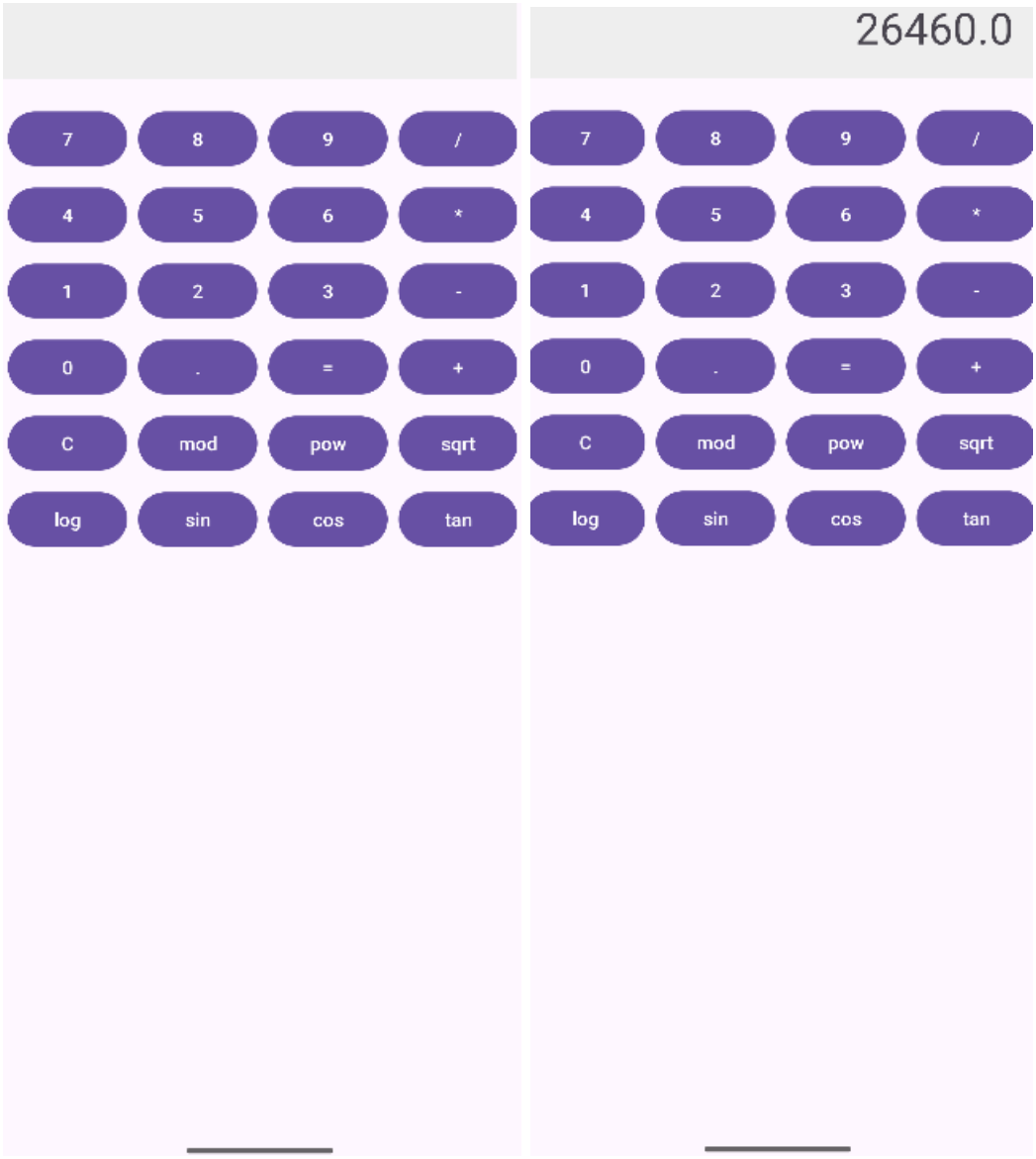
MainActivity.kt

```
package com.example.a2exp23

import android.os.Bundle
import android.view.View
import android.widget.Button
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import kotlin.math.*

class MainActivity : AppCompatActivity() {

    private lateinit var tvInput: TextView
    private var inputText = ""

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
```

```
        tvInput = findViewById(R.id.tvInput)
    }

    fun onDigitClick(view: View) {
        val btn = view as Button
        inputText += btn.text
        tvInput.text = inputText
    }

    fun onOperatorClick(view: View) {
        val btn = view as Button
        inputText += " ${btn.text} "
        tvInput.text = inputText
    }

    fun onFunctionClick(view: View) {
        val btn = view as Button
        inputText += "${btn.text}("
        tvInput.text = inputText
    }

    fun onClearClick(view: View) {
        inputText = ""
        tvInput.text = ""
    }

    fun onEqualClick(view: View) {
        try {
            val result = evaluateExpression(inputText)
            tvInput.text = result.toString()
            inputText = result.toString()
        } catch (e: Exception) {
            tvInput.text = "Error"
        }
    }

    private fun evaluateExpression(expression: String): Double {
        // Basic parsing (for simplicity). You may replace this with a full expression parser.
        val cleaned = expression.replace(" ", "")
        return when {
            cleaned.contains("sqrt(") ->
sqrt(evaluateExpression(cleaned.substringAfter("sqrt(").dropLast(1)))
            cleaned.contains("log(") ->
```

```
ln(evaluateExpression(cleaned.substringAfter("log(").dropLast(1)))
        cleaned.contains("sin(") ->
sin(Math.toRadians(evaluateExpression(cleaned.substringAfter("sin(").dropLast(1))))
        cleaned.contains("cos(") ->
cos(Math.toRadians(evaluateExpression(cleaned.substringAfter("cos(").dropLast(1))))
        cleaned.contains("tan(") ->
tan(Math.toRadians(evaluateExpression(cleaned.substringAfter("tan(").dropLast(1))))
        cleaned.contains("mod") -> {
           val parts = cleaned.split("mod")
           parts[0].toDouble() % parts[1].toDouble()
        }
        cleaned.contains("pow") -> {
           val parts = cleaned.split("pow")
           parts[0].toDouble().pow(parts[1].toDouble())
        }
        cleaned.contains("+") -> {
           val parts = cleaned.split("+")
           parts[0].toDouble() + parts[1].toDouble()
        }
        cleaned.contains("-") -> {
           val parts = cleaned.split("-")
           parts[0].toDouble() - parts[1].toDouble()
        }
        cleaned.contains("*") -> {
           val parts = cleaned.split("*")
           parts[0].toDouble() * parts[1].toDouble()
        }
        cleaned.contains("/") -> {
           val parts = cleaned.split("/")
           parts[0].toDouble() / parts[1].toDouble()
        }
        else -> cleaned.toDouble()
      }
   }
}
```

Output



Result:

The calculator components experiment has been successfully completed