

**Ex. No. : 02**

**Date:**

**Register No.: 221701033**

**Name: Livia Mary Sebastian**

---

## **Simple Calculator**

### **Aim**

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

### ***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 "Finish".

**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.Calculator"
        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"?>
<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">

<Button
    android:id="@+id/button1"
    android:layout_width="59dp"
    android:layout_height="63dp"
    android:layout_marginStart="4dp"
    android:layout_marginTop="252dp"
    android:text="1"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<Button
    android:id="@+id/button4"
    android:layout_width="59dp"
    android:layout_height="63dp"
    android:layout_marginStart="4dp"
    android:layout_marginTop="336dp"
    android:text="4"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<Button
    android:id="@+id/button7"
    android:layout_width="59dp"
    android:layout_height="63dp"
    android:layout_marginStart="4dp"
    android:layout_marginTop="424dp"
    android:text="7"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<Button
    android:id="@+id/button8"
    android:layout_width="59dp"
    android:layout_height="63dp"
    android:layout_marginStart="80dp"
    android:layout_marginTop="424dp"
    android:text="8"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<Button
    android:id="@+id/button0"
    android:layout_width="59dp"
    android:layout_height="63dp"
    android:layout_marginStart="80dp"
    android:layout_marginTop="504dp"
    android:text="0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<Button
    android:id="@+id/buttondot"
    android:layout_width="59dp"
    android:layout_height="63dp"
    android:layout_marginStart="4dp"
    android:layout_marginTop="504dp"
    android:text="."
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<Button
    android:id="@+id/buttonC"
    android:layout_width="76dp"
    android:layout_height="50dp"
    android:layout_marginStart="136dp"
    android:layout_marginTop="652dp"
    android:text="C"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<Button
    android:id="@+id/buttonequals"
    android:layout_width="59dp"
    android:layout_height="63dp"
    android:layout_marginStart="152dp"
    android:layout_marginTop="504dp"
    android:text="="
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<Button
    android:id="@+id/buttonpow"
    android:layout_width="76dp"
    android:layout_height="50dp"
    android:layout_marginStart="248dp"
    android:layout_marginTop="436dp"
    android:text="pow"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<Button
    android:id="@+id/buttonAC"
    android:layout_width="76dp"
    android:layout_height="50dp"
    android:layout_marginStart="132dp"
    android:layout_marginTop="584dp"
    android:text="AC"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<Button
    android:id="@+id/button9"
    android:layout_width="59dp"
    android:layout_height="63dp"
    android:layout_marginStart="152dp"
    android:layout_marginTop="424dp"
    android:text="9"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<Button
    android:id="@+id/button2"
    android:layout_width="59dp"
    android:layout_height="63dp"
    android:layout_marginStart="80dp"
    android:layout_marginTop="252dp"
    android:text="2"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<Button
    android:id="@+id/button5"
    android:layout_width="59dp"
    android:layout_height="63dp"
    android:layout_marginStart="80dp"
    android:layout_marginTop="336dp"
    android:text="5"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<Button
    android:id="@+id/button6"
    android:layout_width="59dp"
    android:layout_height="63dp"
    android:layout_marginStart="152dp"
    android:layout_marginTop="336dp"
    android:text="6"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<Button
    android:id="@+id/button3"
    android:layout_width="59dp"
    android:layout_height="63dp"
    android:layout_marginStart="152dp"
    android:layout_marginTop="252dp"
    android:text="3"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<Button
    android:id="@+id/buttonplus"
    android:layout_width="59dp"
    android:layout_height="63dp"
    android:layout_marginStart="248dp"
    android:layout_marginTop="252dp"
    android:text="+"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<Button
    android:id="@+id/buttonminus"
    android:layout_width="59dp"
    android:layout_height="63dp"
    android:layout_marginStart="324dp"
    android:layout_marginTop="252dp"
    android:text="-"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<Button
    android:id="@+id/buttonmultiply"
    android:layout_width="59dp"
    android:layout_height="63dp"
    android:layout_marginStart="248dp"
    android:layout_marginTop="332dp"
    android:text="x"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<Button
    android:id="@+id/buttonsin"
    android:layout_width="76dp"
    android:layout_height="50dp"
    android:layout_marginStart="332dp"
    android:layout_marginTop="436dp"
    android:text="sin"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<Button
    android:id="@+id/buttoncos"
    android:layout_width="76dp"
    android:layout_height="50dp"
    android:layout_marginStart="332dp"
    android:layout_marginTop="504dp"
    android:text="cos"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<Button
    android:id="@+id/buttonmod"
    android:layout_width="78dp"
    android:layout_height="50dp"
    android:layout_marginStart="248dp"
    android:layout_marginTop="652dp"
    android:text="mod"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<Button
    android:id="@+id/buttonsqrt"
    android:layout_width="78dp"
    android:layout_height="50dp"
    android:layout_marginStart="332dp"
    android:layout_marginTop="576dp"
    android:text="sqrt"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```

< Button
    android:id="@+id/buttontan"
    android:layout_width="76dp"
    android:layout_height="50dp"
    android:layout_marginStart="248dp"
    android:layout_marginTop="504dp"
    android:text="tan"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

< Button
    android:id="@+id/buttonlog"
    android:layout_width="76dp"
    android:layout_height="50dp"
    android:layout_marginStart="248dp"
    android:layout_marginTop="576dp"
    android:text="log"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

< Button
    android:id="@+id/buttondivide"
    android:layout_width="59dp"
    android:layout_height="63dp"
    android:layout_marginStart="324dp"
    android:layout_marginTop="332dp"
    android:text="/"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

< TextView
    android:id="@+id/textView"
    android:layout_width="344dp"
    android:layout_height="193dp"
    android:layout_marginStart="32dp"
    android:layout_marginTop="28dp"
    android:text="0"
    android:textAlignment="viewEnd"
    android:textSize="24dp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>

```



## **MainActivity.kt**

```
package com.example.calculator

import android.os.Bundle
import android.view.View
import android.widget.Button
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import kotlin.math.cos
import kotlin.math.log10
import kotlin.math.sin
import kotlin.math.sqrt
import kotlin.math.tan

class MainActivity : AppCompatActivity() {

    private lateinit var display: TextView
    private var currentInput: String = ""
    private var operator: String? = null
    private var firstNumber: Double? = null
    private var secondNumber: Double? = null

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

        display = findViewById(R.id.textView)

        // Number buttons (dynamically mapped to all numbers 0-9)
        val numberButtons = listOf(
            R.id.button0, R.id.button1, R.id.button2, R.id.button3,
            R.id.button4, R.id.button5, R.id.button6, R.id.button7,
            R.id.button8, R.id.button9,
        )

        // Loop to set click listeners for number buttons dynamically
        numberButtons.forEach { id ->
            findViewById<Button>(id).setOnClickListener {
                val number = (it as Button).text.toString()
                appendNumber(number)
            }
        }
    }
}
```

```

// Operator buttons
val buttonPlus: Button = findViewById(R.id.buttonplus)
buttonPlus.setOnClickListener { setOperator("+") }

val buttonMinus: Button = findViewById(R.id.buttonminus)
buttonMinus.setOnClickListener { setOperator("-") }

val buttonMultiply: Button = findViewById(R.id.buttonmultiply)
buttonMultiply.setOnClickListener { setOperator("*") }

val buttonDivide: Button = findViewById(R.id.buttondivide)
buttonDivide.setOnClickListener { setOperator("/") }

val buttonEquals: Button = findViewById(R.id.buttonequals)
buttonEquals.setOnClickListener { calculateResult() }

val buttonPower: Button = findViewById(R.id.buttonpow)
buttonPower.setOnClickListener { setOperator("pow") }

val buttonSqrt: Button = findViewById(R.id.buttonsqrt)
buttonSqrt.setOnClickListener { setOperator("sqrt") }
//log and trigonometric functions
val buttonSin: Button = findViewById(R.id.buttonsin)
buttonSin.setOnClickListener { setOperator("sin") }

val buttonCos: Button = findViewById(R.id.buttoncos)
buttonCos.setOnClickListener { setOperator("cos") }

val buttonTan: Button = findViewById(R.id.buttontan)
buttonTan.setOnClickListener { setOperator("tan") }

val buttonLog: Button = findViewById(R.id.buttonlog)
buttonLog.setOnClickListener { setOperator("log") }

// Clear button for resetting the calculator
val buttonClear: Button = findViewById(R.id.buttonAC)
buttonClear.setOnClickListener { resetCalculator() }

val buttonClearLast: Button = findViewById(R.id.buttonC)
buttonClearLast.setOnClickListener { clearLastInput() }

val buttonModulus: Button = findViewById(R.id.buttonmod)
buttonModulus.setOnClickListener { setOperator("%") }
}

```

```

// Function to append numbers to the current input
private fun appendNumber(number: String) {
    currentInput += number
    display.text = currentInput
}

// Function to set the operator (like +, -, *, /, ^, √)
private fun setOperator(op: String) {
    if (firstNumber == null) {
        firstNumber = currentInput.toDouble()
        currentInput = ""
    }
    operator = op
    display.text = operator
}

// Function to calculate the result
private fun calculateResult() {
    if (firstNumber != null && operator != null) {
        when (operator) {
            "+" -> {
                secondNumber = currentInput.toDouble()
                val result = firstNumber!! + secondNumber!!
                display.text = result.toString()
            }
            "-" -> {
                secondNumber = currentInput.toDouble()
                val result = firstNumber!! - secondNumber!!
                display.text = result.toString()
            }
            "*" -> {
                secondNumber = currentInput.toDouble()
                val result = firstNumber!! * secondNumber!!
                display.text = result.toString()
            }
            "/" -> {
                secondNumber = currentInput.toDouble()
                val result = if (secondNumber != 0.0) firstNumber!! / secondNumber!! else Double.NaN
                display.text = result.toString()
            }
            "pow" -> { // Power operation (exponentiation)
                secondNumber = currentInput.toDouble()
                val result = Math.pow(firstNumber!!, secondNumber!!)
                display.text = result.toString()
            }
        }
    }
}

```

```

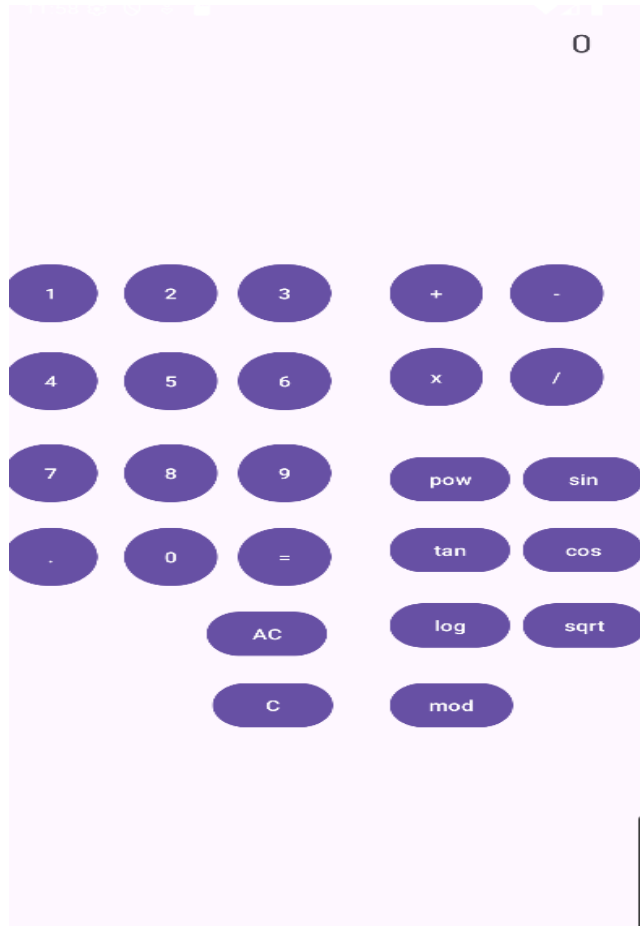
"sqrt" -> { // Square Root operation
    val result = sqrt(firstNumber!!)
    display.text = result.toString()
}
"sin" -> { // Sine operation (in radians)
    val result = sin(Math.toRadians(firstNumber!!))
    display.text = result.toString()
}
"cos" -> { // Cosine operation (in radians)
    val result = cos(Math.toRadians(firstNumber!!))
    display.text = result.toString()
}
"tan" -> { // Tangent operation (in radians)
    val result = tan(Math.toRadians(firstNumber!!))
    display.text = result.toString()
}
"log" -> { // Logarithm operation (base 10)
    val result = log10(firstNumber!!)
    display.text = result.toString()
}
"%" -> { // Modulus operation
    secondNumber = currentInput.toDouble()
    val result = firstNumber!! % secondNumber!!
    display.text = result.toString()
}
else -> display.text = "Error"
}
//resetCalculator() // Reset after showing the result
}

// Function to reset calculator for next calculation
private fun resetCalculator() {
    firstNumber = null
    secondNumber = null
    currentInput = ""
    operator = null
    display.text = "0" // Reset the display text to "0"
}

// Function to clear the last digit or value from the current input
private fun clearLastInput() {
    // Clears the last character from the current input (from right to left)
    if (currentInput.isNotEmpty()) {
        currentInput = currentInput.dropLast(1) // Remove last character
        // Update the display to reflect the current input (or "0" if input is empty)
        display.text = if (currentInput.isNotEmpty()) currentInput else "0"
    }
}

```

## Output



## Result:

The experiment was conducted successfully

