Ex. No. : 02 **Date: 24.02.2025**

Register No.: 221701020 **Name: Hemalatha R**

ScientificCalculator

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 "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.Ex2"
 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://sche

mas.android.com/apk/res/a

ndroid"
xmlns:app="http://schemas.

android.com/apk/res-auto"
```

```
android:layout_width="mat
ch_parent"

android:layout_height="ma
tch_parent"

android:padding="16dp">

 <LinearLayout

android:layout_width="mat
ch_parent"

android:layout_height="wr
ap_content"

android:orientation="vertic
al"

android:gravity="center_ho
rizontal">

 <!-- Fixed: Added
layout_width and
layout_height -->

<com.google.android.mater
ial.textfield.TextInputLayout
```

```
android:layout_width="mat
ch_parent"

android:layout_height="wr
ap_content"

android:minHeight="56dp"

app:boxBackgroundMode="
outline"

app:boxStrokeColor="@and
roid:color/black">


<com.google.android.mater
ial.textfield.TextInputEdit
Text

android:id="@+id/etInput"

android:layout_width="mat
ch_parent"

android:layout_height="wr
ap_content"

android:hint="Enter
number(s) (e.g. 5 3)"

android:inputType="text"
/>
```

```
</com.google.android.mater
ial.textfield.TextInputLayo
ut>

 <!-- Trigonometric
Buttons -->
 <LinearLayout

android:layout_width="mat
ch_parent"

android:layout_height="wr
ap_content"

android:orientation="horiz
ontal"

android:gravity="center"

android:paddingTop="16dp
">

 <Button
android:id="@+id/btnSin"
android:layout_width="wra
p_content"
android:layout_height="wr
ap_content"
android:text="sin" />
 <Button
android:id="@+id/btnCos"
android:layout_width="wra
p_content"
```

```
android:layout_height="wr
ap_content"
android:text="cos" />
 <Button
android:id="@+id/btnTan"
android:layout_width="wra
p_content"
android:layout_height="wr
ap_content"
android:text="tan" />
 </LinearLayout>

 <!-- Arithmetic
Buttons -->
 <LinearLayout

android:layout_width="mat
ch_parent"

android:layout_height="wr
ap_content"

android:orientation="horiz
ontal"

android:gravity="center"

android:paddingTop="8dp"
>

 <Button
android:id="@+id/btnAdd"
android:layout_width="wra
```

```
p_content"
android:layout_height="wr
ap_content"
android:text="+" />
 <Button
android:id="@+id/btnSubtr
act"
android:layout_width="wra
p_content"
android:layout_height="wr
ap_content" android:text="-
" />
 <Button
android:id="@+id/btnMulti
ply"
android:layout_width="wra
p_content"
android:layout_height="wr
ap_content"
android:text="*" />
 <Button
android:id="@+id/btnDivid
e"
android:layout_width="wra
p_content"
android:layout_height="wr
ap_content"
android:text="/" />
 </LinearLayout>

 <!-- Scientific Buttons
-->
 <LinearLayout
```

```
android:layout_width="mat
ch_parent"

android:layout_height="wr
ap_content"

android:orientation="horiz
ontal"

android:gravity="center"

android:paddingTop="8dp"
>

 <Button
android:id="@+id/btnSqrt"
android:layout_width="wra
p_content"
android:layout_height="wr
ap_content"
android:text="√" />
 <Button
android:id="@+id/btnPow"
android:layout_width="wra
p_content"
android:layout_height="wr
ap_content"
android:text="^" />
 <Button
android:id="@+id/btnLog"
android:layout_width="wra
p_content"
```

```
android:layout_height="wr
ap_content"
android:text="In" />
 <Button
android:id="@+id/btnMod"
android:layout_width="wra
p_content"
android:layout_height="wr
ap_content"
android:text="%" />
 </LinearLayout>

 <!-- Result Display -->
 <TextView

android:id="@+id/tvResult"

android:layout_width="mat
ch_parent"

android:layout_height="wr
ap_content"

android:text="Result:"

android:textSize="24sp"

android:gravity="center"

android:paddingTop="24dp
" />
```

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

MainActivity.kt

```
 package com.example.ex2

import
 androidx.appcompat.app.A
ppCompatActivity
import android.os.Bundle
import android.widget.*
import kotlin.math.*

class MainActivity :
AppCompatActivity() {

 override fun
onCreate(savedInstanceSta
te: Bundle?) {

super.onCreate(savedInsta
nceState)

setContentView(R.layout.a
ctivity_main)

 val etInput =
findViewById<EditText>(R
.id.etInput)
 val tvResult =
findViewById<TextView>(
```

```
R.id.tvResult)

 val btnAdd =
findViewById<Button>(R.i
d.btnAdd)
 val btnSubtract =
findViewById<Button>(R.i
d.btnSubtract)
 val btnMultiply =
findViewById<Button>(R.i
d.btnMultiply)
 val btnDivide =
findViewById<Button>(R.i
d.btnDivide)
 val btnSin =
findViewById<Button>(R.i
d.btnSin)
 val btnCos =
findViewById<Button>(R.i
d.btnCos)
 val btnTan =
findViewById<Button>(R.i
d.btnTan)
 val btnSqrt =
findViewById<Button>(R.i
d.btnSqrt)
 val btnPow =
findViewById<Button>(R.i
d.btnPow)
 val btnLog =
findViewById<Button>(R.i
d.btnLog)
 val btnMod =
```

```
findViewById<Button>(R.i
d.btnMod)


btnAdd.setOnClickListener
{

calculateTwoInputs("+",
etInput, tvResult)
 }


btnSubtract.setOnClickLis
tener {

calculateTwoInputs("-",
etInput, tvResult)
 }


btnMultiply.setOnClickList
ener {

calculateTwoInputs("*",
etInput, tvResult)
 }


btnDivide.setOnClickListe
ner {

calculateTwoInputs("/",
etInput, tvResult)
```

```
}


btnSin.setOnClickListener
{
 val input =
etInput.text.toString().toDo
ubleOrNull()
 input?.let {
 val result =
sin(Math.toRadians(it))
 tvResult.text =
"Result: $result"
 } ?: showError()
 }


btnCos.setOnClickListener
{
 val input =
etInput.text.toString().toDo
ubleOrNull()
 input?.let {
 val result =
cos(Math.toRadians(it))
 tvResult.text =
"Result: $result"
 } ?: showError()
 }

btnTan.setOnClickListener
{
```

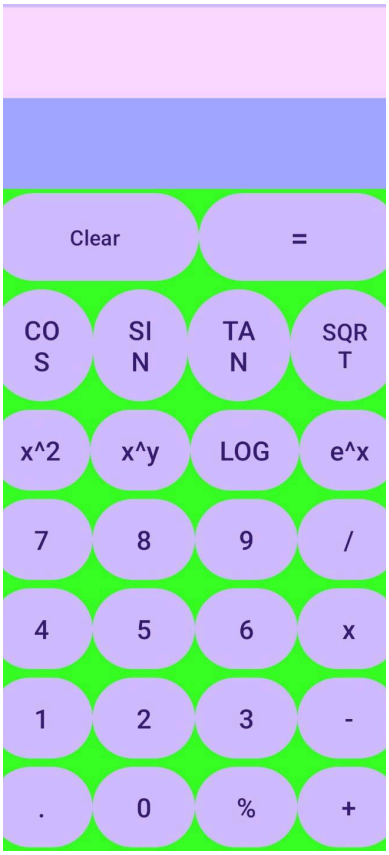
```
val input =
etInput.text.toString().toDo
ubleOrNull()
 input?.let {
 val result =
tan(Math.toRadians(it))
 tvResult.text =
"Result: $result"
 } ?: showError()
 }


btnSqrt.setOnClickListene
r {
 val input =
etInput.text.toString().toDo
ubleOrNull()
 input?.let {
 if (it >= 0) {
 val result =
sqrt(it)
 tvResult.text =
"Result: $result"
 } else {
 tvResult.text =
"Error: Negative number!"
 }
 } ?: showError()
 }


btnPow.setOnClickListener
{
```

```
calculateTwoInputs("^",
etInput, tvResult)
 }


btnLog.setOnClickListener
{
 val input =
etInput.text.toString().toDo
ubleOrNull()
 input?.let {
 if (it > 0) {
 val result =
ln(it)
 tvResult.text =
"Result: $result"
 } else {
 tvResult.text =
"Error: Input must be > 0!"
 }
 } ?: showError()
 }


btnMod.setOnClickListene
r {

calculateTwoInputs("%",
etInput, tvResult)
 }
 }
```

```
 private fun
calculateTwoInputs(operati
on: String, etInput:
EditText, tvResult:
TextView) {
 val inputText =
etInput.text.toString()
 val numbers =
inputText.split(" ")
 if (numbers.size != 2) {
 tvResult.text =
"Enter two numbers
separated by space."
 return
 }

 val a =
numbers[0].toDoubleOrNul
l()
 val b =
numbers[1].toDoubleOrNul
l()

 if (a == null || b ==
null) {
 showError()
 return
 }

 val result = when
(operation) {
 "+" -> a + b
 "-" -> a - b
```

```
 "*" -> a * b
 "/" -> if (b != 0.0) a /
b else "Error: Divide by
zero"
 "^" -> a.pow(b)
 "%" -> a % b
 else -> "Unknown
operation"
 }

 tvResult.text =
"Result: $result"
 }

 private fun showError() {
 Toast.makeText(this,
"Invalid input!",
Toast.LENGTH_SHORT).s
how()
 }
}
```

Output :



Result:

The Application was developed using Kotlin in Android Studio.