**Step 1: Set Up Your Project**

1. Open Android Studio and create a new project.
2. Choose an Empty Activity template and Kotlin as the programming language.

**Step 2: Design the Layout**

Design your calculator layout (**activity\_main.xml**) with buttons representing numbers, operators, and other controls.

#### activity\_main.xml

<Button

android:id="@+id/btn\_zero"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="0"

android:layout\_weight="1"/>

### Step 3: Set Up Button Click Listeners

In your MainActivity, set up onClickListeners for the calculator buttons to handle user input and perform calculations.

#### MainActivity.kt

import android.os.Bundle

import android.view.View

import android.widget.Button

import android.widget.TextView

import androidx.appcompat.app.AppCompatActivity

import javax.script.ScriptEngineManager

class MainActivity : AppCompatActivity() {

private lateinit var resultTextView: TextView

private var inputString = ""

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

resultTextView = findViewById(R.id.resultTextView)

}

fun onNumberClick(view: View) {

val button = view as Button

inputString += button.text

resultTextView.text = inputString

}

fun onOperatorClick(view: View) {

inputString += " ${(view as Button).text} "

resultTextView.text = inputString

}

fun onClearClick(view: View) {

inputString = ""

resultTextView.text = ""

}

fun onCalculateClick(view: View) {

val result = try {

ScriptEngineManager().getEngineByName("rhino").eval(inputString)

} catch (e: Exception) {

"Error"

}

resultTextView.text = result.toString()

}

}

### Step 4: Implement Calculation Logic

Implement the calculation logic in the **onCalculateClick** method to evaluate the expression entered by the user and display the result.

### Step 5: Test Your App

Run your app on a device or emulator, and test the functionality of your calculator by entering numbers and operators.