Experiment 5: Basic Calculator App

1. Objective

Create a simple calculator that can perform basic arithmetic operations like addition, subtraction, multiplication, and division. This experiment introduces more complex UI layouts, event handling, and performing arithmetic operations based on user input.

2. Steps to Complete the Experiment

1. Design the Calculator UI:

   Start by creating a TextView that will display inputs and results.

   Arrange Button elements in a grid-like format for the digits (0-9) and operations (+, -, *, /, =, and C for clear).

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/main"
    tools:context=".MainActivity">

    <!-- Row 1: 7, 8, 9, / -->

    <TextView
```

```
    android:id="@+id/tvDisplay"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="72dp"
    android:background="@android:color/darker_gray"
    android:padding="16dp"
    android:text="0"
    android:textAlignment="textEnd"
    android:textSize="32sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<Button
    android:id="@+id/btnSeven"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="7"
    app:layout_constraintTop_toBottomOf="@id/tvDisplay"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toStartOf="@id/btnEight"
    app:layout_constraintHorizontal_chainStyle="spread"
    android:onClick="numberEvent"/>

<Button
```

```
    android:id="@+id/btnEight"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="8"
    app:layout_constraintTop_toTopOf="@id/btnSeven"
    app:layout_constraintStart_toEndOf="@id/btnSeven"
    app:layout_constraintEnd_toStartOf="@id/btnNine"
    android:onClick="numberEvent"/>

<Button
    android:id="@+id/btnNine"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="9"
    app:layout_constraintTop_toTopOf="@id/btnEight"
    app:layout_constraintStart_toEndOf="@id/btnEight"
    app:layout_constraintEnd_toStartOf="@id/btnDivide"
    android:onClick="numberEvent"/>

<Button
    android:id="@+id/btnDivide"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="/"
    app:layout_constraintTop_toTopOf="@id/btnNine"
    app:layout_constraintStart_toEndOf="@id/btnNine"
```

```
    app:layout_constraintEnd_toEndOf="parent"
    android:onClick="operationEvent"/>

<!-- Row 2: 4, 5, 6, * -->
<Button
    android:id="@+id/btnFour"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="4"
    app:layout_constraintTop_toBottomOf="@id/btnSeven"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toStartOf="@id/btnEight"
    app:layout_constraintHorizontal_chainStyle="spread"
    android:onClick="numberEvent"/>

<Button
    android:id="@+id/btnFive"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="5"
    app:layout_constraintTop_toTopOf="@id/btnFour"
    app:layout_constraintStart_toEndOf="@id/btnFour"
    app:layout_constraintEnd_toStartOf="@id/btnSix"
    android:onClick="numberEvent"/>

<Button
```

```
    android:id="@+id/btnSix"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="6"
    app:layout_constraintTop_toTopOf="@id/btnFive"
    app:layout_constraintStart_toEndOf="@id/btnFive"
    app:layout_constraintEnd_toStartOf="@id/btnMultiply"
    android:onClick="numberEvent"/>

<Button
    android:id="@+id/btnMultiply"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="*"
    app:layout_constraintTop_toTopOf="@id/btnSix"
    app:layout_constraintStart_toEndOf="@id/btnSix"
    app:layout_constraintEnd_toEndOf="parent"
    android:onClick="operationEvent"/>

<!-- Row 3: 1, 2, 3, - -->
<Button
    android:id="@+id/btnOne"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="1"
    app:layout_constraintTop_toBottomOf="@id/btnFour"
```

```
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toStartOf="@id/btnTwo"
    app:layout_constraintHorizontal_chainStyle="spread"
    android:onClick="numberEvent"/>

<Button
    android:id="@+id/btnTwo"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="2"
    app:layout_constraintTop_toTopOf="@id/btnOne"
    app:layout_constraintStart_toEndOf="@id/btnOne"
    app:layout_constraintEnd_toStartOf="@id/btnThree"
    android:onClick="numberEvent"/>

<Button
    android:id="@+id/btnThree"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="3"
    app:layout_constraintTop_toTopOf="@id/btnTwo"
    app:layout_constraintStart_toEndOf="@id/btnTwo"
    app:layout_constraintEnd_toStartOf="@id/btnMinus"
    android:onClick="numberEvent"/>

<Button
```

```
    android:id="@+id/btnMinus"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="-"
    app:layout_constraintTop_toTopOf="@id/btnThree"
    app:layout_constraintStart_toEndOf="@id/btnThree"
    app:layout_constraintEnd_toEndOf="parent"
    android:onClick="operationEvent"/>

<!-- Row 4: 0, C, =, + -->
<Button
    android:id="@+id/btnZero"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="0"
    app:layout_constraintTop_toBottomOf="@id/btnOne"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toStartOf="@id/btnClear"
    app:layout_constraintHorizontal_chainStyle="spread"
    android:onClick="numberEvent"/>

<Button
    android:id="@+id/btnClear"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="C"
```

```
        app:layout_constraintTop_toTopOf="@id/btnZero"
        app:layout_constraintStart_toEndOf="@id/btnZero"
        app:layout_constraintEnd_toStartOf="@id/btnEquals"
        android:onClick="clearEvent"/>

    <Button
        android:id="@+id/btnEquals"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:text="="
        app:layout_constraintTop_toTopOf="@id/btnClear"
        app:layout_constraintStart_toEndOf="@id/btnClear"
        app:layout_constraintEnd_toStartOf="@id/btnAdd"
        android:onClick="equalEvent"/>

    <Button
        android:id="@+id/btnAdd"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:text="+"
        app:layout_constraintTop_toTopOf="@id/btnEquals"
        app:layout_constraintStart_toEndOf="@id/btnEquals"
        app:layout_constraintEnd_toEndOf="parent"
        android:onClick="operationEvent"/>

</androidx.constraintlayout.widget.ConstraintLayout>
```

2. Implement Button Functionality:

   Set up onClick listeners for each button to handle user interactions.

   For digit buttons, append the corresponding number to the display.

   For operation buttons, store the current number and the selected operation.

   The equals button should perform the calculation based on the stored number, operation, and the current number on the display.

   The clear button resets everything to the initial state.

3. Perform Calculations:

   Implement the logic for performing arithmetic operations based on user inputs.

   Update the display with the result when the user presses the equals button.

4. Test the Application:

   Run the application on an emulator or a real device.

   Test each functionality: digit input, operations, and clearing the result.

```
package com.yourpackage.name; // Change this to your actual

import android.os.Bundle;

import android.view.View;

import android.widget.TextView;


import androidx.activity.EdgeToEdge;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.graphics.Insets;

import androidx.core.view.ViewCompat;

import androidx.core.view.WindowInsetsCompat;


public class MainActivity extends AppCompatActivity {
```

```
    TextView tvDisplay;
    private double firstNumber = 0;
    private String operation = null;
    private boolean isNewOperation = true;


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

ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.ma
in), (v, insets) -> {
            Insets                systemBars                =
insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left,      systemBars.top,
systemBars.right, systemBars.bottom);
            return insets;
        });

        tvDisplay = findViewById(R.id.tvDisplay);
    }

    public void numberEvent(View view) {
```

```
if (isNewOperation) {
    tvDisplay.setText("");
}
isNewOperation = false;
String number = tvDisplay.getText().toString();
int id = view.getId();
if (id == R.id.btnZero) {
    number += "0";
} else if (id == R.id.btnOne) {
    number += "1";
} else if (id == R.id.btnTwo) {
    number += "2";
} else if (id == R.id.btnThree) {
    number += "3";
} else if (id == R.id.btnFour) {
    number += "4";
} else if (id == R.id.btnFive) {
    number += "5";
} else if (id == R.id.btnSix) {
    number += "6";
} else if (id == R.id.btnSeven) {
    number += "7";
} else if (id == R.id.btnEight) {
    number += "8";
} else if (id == R.id.btnNine) {
    number += "9";
```

```
        }
        tvDisplay.setText(number);
    }

    public void operationEvent(View view) {
        isNewOperation = true;
        firstNumber =
Double.parseDouble(tvDisplay.getText().toString());
        int id = view.getId();
        if (id == R.id.btnAdd) {
            operation = "+";
        } else if (id == R.id.btnMinus) {
            operation = "-";
        } else if (id == R.id.btnMultiply) {
            operation = "*";
        } else if (id == R.id.btnDivide) {
            operation = "/";
        }
    }

    public void equalEvent(View view) {
        if (operation == null) {
            tvDisplay.setText("No operation set");
            isNewOperation = true;
```

```
            return;
        }
        String newNumber = tvDisplay.getText().toString();
        double result = 0.0;
        switch (operation) {
            case "+":
                result = firstNumber +
Double.parseDouble(newNumber);
                break;
            case "-":
                result = firstNumber -
Double.parseDouble(newNumber);
                break;
            case "*":
                result = firstNumber *
Double.parseDouble(newNumber);
                break;
            case "/":
                if (Double.parseDouble(newNumber) != 0) {
                    result = firstNumber /
Double.parseDouble(newNumber);
                } else {
                    tvDisplay.setText("Error");
                    isNewOperation = true;
                    return;
                }
```

```
                break;
        }
        tvDisplay.setText(String.valueOf(result));
        isNewOperation = true;
    }

    public void clearEvent(View view) {
        tvDisplay.setText("0");
        isNewOperation = true;
        firstNumber = 0;
        operation = null;
    }

}
```

Ensure you replace com.yourpackage.name with your actual package name at the top of the file.

Double-check that all button IDs in the numberEvent and operationEvent methods match those in your activity_main.xml.

The equalEvent method performs the calculation based on the selected operation and updates the display with the result.

The clearEvent method resets the calculator to its initial state.

In MainActivity.java, ensure you have methods corresponding to the android:onClick attributes set in your layout buttons. The provided snippets include the structure and logic, but you'll need to complete it by adding all digit and operation cases.