

**FACULTY OF ENGINEERING AND  
TECHNOLOGY BACHELOR OF TECHNOLOGY**

**MOBILE APP DEVELOPMENT LABORATORY  
(303105380)**

**VI SEMESTER**

**CSE Department**



**Laboratory Manual  
Session 2025-26**

# CERTIFICATE

This is to certify that

Mr / Ms \_\_\_\_\_ with enrollment no. \_\_\_\_\_ has successfully completed his/her laboratory experiments in the **Mobile App Development** (303105380) from the department of **COMPUTER SCIENCE & ENGINEERING** during the academic year 2025-2026.



Date of Submission:.....

Staff In charge:.....

Head Of Department:.....



## TABLE OF CONTENT

Sr. No	Experiment Title	Page No		Date of Start	Date of Completion	Marks (out of 10)	Sign
		F r o m	T o				
1.	Create a "Hello World" application: Display "Hello World" at the center of the screen, both on the Android emulator and an actual Android device.						
2.	Build an app to showcase Android lifecycle phases: Develop an app that demonstrates various Android lifecycle stages (onCreate, onStart, onResume, etc.).						
3.	Create an app with two activities: The first activity should contain an EditText and a "Send" button. When the button is clicked, use an explicit intent to send the text from EditText to a second activity and display it in a TextView						
4.	Create an app with explicit intent: The first activity should have an EditText and a "Send" button. On button click, use an implicit intent with the "SEND" action, allowing the user to select an app from an app chooser to handle the intent and display the text.						
5.	Build a basic calculator app: Create an app that performs basic arithmetic operations (addition, subtraction, multiplication, and division) on numbers						
6.	Create a Spinner-based app: Develop an app with a spinner populated from the res/values/strings.xml resource. When the spinner value changes, the corresponding image from the res/drawable directory should be displayed						
7.	Create a discount calculator app: Use a RadioGroup with three radio buttons for 10%, 15%, and 20% discounts on a shopping bill. The user can enter the bill amount in an EditText, and the selected discount will be calculated and displayed in a TextView.						
8.	Create an app with a course selection RadioButton group: Display a list of college courses with a RadioButtongroup. When a course is selected, the corresponding TIC (Total Instructional Credit) should be shown in a TextView.						

<b>9.</b>	Create a shopping list app using checkboxes: Build an app with checkboxes for shopping list items. As items are checked off, the selected items should be displayed in a TextView.					
<b>10.</b>	Create a login and registration app: Develop a login application that verifies the username and password. Include a registration page for new users. Upon successful login, show a "Welcome User" pop-up message					
<b>11.</b>	Create a login app with navigation to another activity: The login screen should verify the username and password. After successful login, navigate to a new activity that displays a "Welcome User" message in a TextView and a "Logout" button. On clicking "Logout," show a confirmation dialog with "OK" and "Cancel" buttons. "OK" should return to the login screen, while "Cancel" should keep the user on the current activity.					
<b>12.</b>	Create an app with a menu: Implement a menu with five options. The selected option should be displayed in a TextView.					
<b>13.</b>	Build an app using LinearLayout: Create a simple app that uses LinearLayout. It should take the contents of a predefined TextView, convert it to uppercase on button click, and display it in an EditText. Additionally, create an app that responds to key events in the EditText without needing a button press					
<b>14.</b>	Create an app with TableLayout and custom styles: Use a TableLayout with a TextView, EditText, and buttons. Also, create a custom styles.xml in the res/values directory to style the TextView.					
<b>15.</b>	Create an app with SQLite database operations: Build an app that allows the user to perform CRUD operations (Create, Read, Update, Delete) with an SQLite database					
<b>16.</b>	Create an app with three vertically aligned buttons: Develop an app with three buttons arranged vertically. When any button is selected, the screen color should change accordingly.					



## PRACTICAL NO: 01

**AIM:** To develop an Android application that displays “Hello World” at the center of the screen.

**THEORY:** A basic Android app uses TextView inside a layout. The setContentView() method loads the UI on the screen. The gravity attribute is used to center the text.

### CODE SNIPPET:

*activity\_main.xml code:*

```
<?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"
    tools:context=".MainActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="HelloWorld"
        android:textSize="30dp"
        android:textStyle="bold"
        android:layout_marginTop="280dp"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```



### MainActivity.java code:

```
package com.example.pract_4_hello_world;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

### OUTPUT:

The screenshot shows the Android Studio interface. On the left, the project structure is visible with files like activity\_main.xml, MainActivity.java, and various resource files. The main area displays the Java code for MainActivity.java, which contains the onCreate method. On the right, the emulator window shows the text "Hello World" centered on the screen.

```
<?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"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World"
        android:textSize="30dp"
        android:textStyle="bold"
        android:layout_marginTop="280dp"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"/>

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



## PRACTICAL NO: 2

**AIM:** To develop an app that displays messages for each lifecycle method.

**THEORY:** Android Activity lifecycle describes the states an activity goes through from creation to destruction. Understanding lifecycle is essential for correct resource management, persisting UI state, and responding to configuration changes.

Common lifecycle callbacks:

- `onCreate(Bundle savedInstanceState)` — called once when activity is first created. Initialize UI, restore state.
- `onStart()` — activity becomes visible to the user (but not yet interactive).
- `onResume()` — activity starts interacting with the user (at top of the stack). Register listeners, start animations.
- `onPause()` — activity is partially obscured (e.g., another semi-transparent activity on top) or user is leaving; quick operations, persist lightweight data.
- `onStop()` — activity no longer visible. Release resources that aren't needed while visible.
- `onRestart()` — called when activity moves from stopped to started again.
- `onDestroy()` — final cleanup before activity is removed from memory (may not be called if system kills process).

Common user actions and lifecycle effects:

- Rotate device (configuration change): default behavior destroys and recreates activity, so you will see `onPause → onStop → onDestroy → onCreate → onStart → onResume`. (You can override this with `configChanges`, but in this lab we keep default.)
- Tap Home: `onPause → onStop`. Returning: `onRestart → onStart → onResume`.
- Open new activity: current activity `onPause` (maybe `onStop` if fully hidden); new activity `onCreate → onStart → onResume`.
- Press Back on an activity: `onPause → onStop → onDestroy`.

This app logs lifecycle callbacks to:

1. Android Logcat (`Log.d`) — useful for debugging.
2. Toasts — brief on-screen messages for immediate observation.
3. A scrollable TextView that accumulates lifecycle events for demonstration and printing.

### CODE SNIPPET:

#### activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/rootLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="12dp">
```

```
<TextView
    android:id="@+id/titleTxt"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Lifecycle Demo - MainActivity"
    android:textSize="20sp"
    android:gravity="center"
    android:padding="8dp" />
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginTop="8dp"
    android:gravity="center">
```

```
    <Button
        android:id="@+id	btnOpenSecond"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Open Second Activity" />
```

```
    <Button
        android:id="@+id	btnClear"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Clear Log"
        android:layout_marginStart="12dp" />
</LinearLayout>
```

```
<TextView
    android:id="@+id/logLabel"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Lifecycle Events:"
    android:layout_marginTop="12dp"
    android:textStyle="bold" />
```

```
<ScrollView
    android:id="@+id/scrollView"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:fillViewport="true"
    android:background="#F6F6F6"
    android:padding="8dp">
```

```
    <TextView
        android:id="@+id	logTextView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text=""
        android:textSize="14sp" />
```

```
</ScrollView>
```

```
</LinearLayout>
```

### **activity\_second.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/rootSecond"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="12dp">

    <TextView
        android:id="@+id/titleSecond"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Lifecycle Demo - SecondActivity"
        android:textSize="20sp"
        android:gravity="center"
        android:padding="8dp" />

    <Button
        android:id="@+id	btnFinish"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Finish (Back)" />

    <TextView
        android:id="@+id/logTextViewSecond"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Events will appear here..."
        android:layout_marginTop="12dp" />
</LinearLayout>
```

### **MainActivity.java**

```
package com.example.lifecycle;

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.ScrollView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
```

```
public class MainActivity extends AppCompatActivity {  
    private static final String TAG = "MainActivityLifecycle";  
    private TextView logTextView;  
    private ScrollView scrollView;  
    private Button btnOpenSecond, btnClear;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        logTextView = findViewById(R.id.logTextView);  
        scrollView = findViewById(R.id.scrollView);  
        btnOpenSecond = findViewById(R.id.btnOpenSecond);  
        btnClear = findViewById(R.id.btnClear);  
  
        btnOpenSecond.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View view) {  
                Intent i = new Intent(MainActivity.this, SecondActivity.class);  
                startActivity(i);  
            }  
        });  
  
        btnClear.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View view) {  
                logTextView.setText("");  
            }  
        });  
  
        logAndShow("onCreate");  
    }  
  
    @Override  
    protected void onStart() {  
        super.onStart();  
        logAndShow("onStart");  
    }  
  
    @Override  
    protected void onResume() {  
        super.onResume();  
        logAndShow("onResume");  
    }  
  
    @Override  
    protected void onPause() {  
        super.onPause();  
        logAndShow("onPause");  
    }  
  
    @Override  
    protected void onStop() {
```

```
super.onStop();
logAndShow("onStop");
}

@Override
protected void onRestart() {
    super.onRestart();
    logAndShow("onRestart");
}

@Override
protected void onDestroy() {
    super.onDestroy();
    logAndShow("onDestroy");
}

private void logAndShow(String methodName) {
    String message = methodName + " called at " + System.currentTimeMillis();
    // log to Logcat
    Log.d(TAG, message);
    // show Toast
    Toast.makeText(this, methodName, Toast.LENGTH_SHORT).show();
    // append to on-screen log
    logTextView.append(methodName + "\n");
    // scroll to bottom so latest event is visible
    scrollView.post(new Runnable() {
        @Override
        public void run() {
            scrollView.fullScroll(View.FOCUS_DOWN);
        }
    });
}
}
```

## SecondActivity.java

```
package com.example.lifecycle;

import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class SecondActivity extends AppCompatActivity {
    private static final String TAG = "SecondActivityLifecycle";
    private TextView logTextViewSecond;
    private Button btnFinish;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);
        logTextViewSecond = findViewById(R.id.logTextViewSecond);
        btnFinish = findViewById(R.id.btnExit);

        btnFinish.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                finish(); // ends this activity and returns to MainActivity
            }
        });

        logAndShow("onCreate");
    }

    @Override
    protected void onStart() {
        super.onStart();
        logAndShow("onStart");
    }

    @Override
    protected void onResume() {
        super.onResume();
        logAndShow("onResume");
    }

    @Override
    protected void onPause() {
        super.onPause();
        logAndShow("onPause");
    }
}
```

```
@Override
protected void onStop() {
    super.onStop();
    logAndShow("onStop");
}

@Override
protected void onRestart() {
    super.onRestart();
    logAndShow("onRestart");
}

@Override
protected void onDestroy() {
    super.onDestroy();
    logAndShow("onDestroy");
}

private void logAndShow(String methodName) {
    String message = methodName + " called";
    Log.d(TAG, message);
    Toast.makeText(this, methodName, Toast.LENGTH_SHORT).show();
    // Append to TextView (simple)
    logTextViewSecond.append(methodName + "\n");
}
}
```

**OUTPUT :**



## PRACTICAL NO: 03

**AIM:** To develop an Android application with two activities where the first activity takes user input from an EditText and sends it to the second activity using an explicit intent, and the second activity displays the received message in a TextView.

### CODE SNIPPET:

#### activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="20dp">

    <EditText
        android:padding="50dp"
        android:layout_margin="@+id/app_bar"
        android:id="@+id/etMessage"
        android:layout_width="match_parent"
        android:layout_height="150dp"
        android:hint="Enter Your Message"
        android:gravity="top"
        android:inputType="textMultiLine"/>

    <Button
        android:id="@+id/btnSend"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="Send" />

</LinearLayout>
```

#### activity\_second.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:padding="20dp">
```

```
<TextView
    android:id="@+id/tvMessage"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="20sp"
    android:textStyle="bold"/>

<Button
    android:id="@+id	btnGoBack"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="30dp"
    android:text="Go Back to First Screen"
    android:textSize="16sp"/>

</LinearLayout>
```

### MainActivity.java

```
package com.example.intentdemo;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    EditText etMessage;
    Button btnSend;

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

        etMessage = findViewById(R.id.etMessage);
        btnSend = findViewById(R.id.btnSend);

        btnSend.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
```

```
String message = etMessage.getText().toString();

Intent intent = new Intent(MainActivity.this, SecondActivity.class);
intent.putExtra("msg", message);
startActivity(intent);
}
}
);
}
```

### **SecondActivity.java**

```
import android.content.Intent;
import android.os.Bundle;
import android.widget.Button;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

public class SecondActivity extends AppCompatActivity {

    TextView tvMessage;
    Button btnGoBack;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);

        tvMessage = findViewById(R.id.tvMessage);
        btnGoBack = findViewById(R.id.btnGoBack);

        // Get data from Intent
        String receivedMessage = getIntent().getStringExtra("msg");

        tvMessage.setText(receivedMessage);

        btnGoBack.setOnClickListener(v -> {
            finish(); // Back to MainActivity
        });
    }
}
```

### **OUTPUT:**



## PRACTICAL NO: 04

**AIM:** To create an Android application where the first activity contains an EditText and a Send button. When the button is clicked, an implicit intent with the **SEND** action is used to open an app chooser so the user can select any supported application (Messages, Email, WhatsApp, Notes, etc.) to display/share the entered text.

### CODE SNIPPET:

#### **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"
    android:padding="16dp"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/editTextMessage"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="100dp"
        android:hint="Enter your message here"
        android:inputType="textMultiLine"
        android:minLines="5"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/buttonSend"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="24dp"
        android:text="Send"
        android:textSize="16sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
```

```
    app:layout_constraintTop_toBottomOf="@+id/editTextMessage" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>
```

## MainActivity.java

```
package com.example.implicitintentapp;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
  
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 {  
  
    private EditText editTextMessage;  
    private Button buttonSend;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        EdgeToEdge.enable(this);  
        setContentView(R.layout.activity_main);  
  
        editTextMessage = findViewById(R.id.editTextMessage);  
        buttonSend = findViewById(R.id.buttonSend);  
  
        buttonSend.setOnClickListener(new View.OnClickListener() {  
                    @Override  
                    public void onClick(View v) {  
                            sendMessage();  
                    }  
            });
```

```
    }

    private void sendMessage() {
        String message = editTextMessage.getText().toString().trim();

        if (message.isEmpty()) {
            Toast.makeText(this, "Please enter a message",
Toast.LENGTH_SHORT).show();
            return;
        }

        Intent sendIntent = new Intent();

        sendIntent.setAction(Intent.ACTION_SEND).putExtra(Intent.EXTRA_
TEXT, message).setType("text/plain");
        Intent chooser = Intent.createChooser(sendIntent, "Send
message via:");

        if (sendIntent.resolveActivity(getApplicationContext()) != null) {
            startActivity(chooser);
        } else {
            Toast.makeText(this, "No apps available to handle this action",
Toast.LENGTH_SHORT).show();
        }
    }
}
```

## OUTPUT:



## PRACTICAL NO : 05

**AIM:** To create an Android application that performs basic arithmetic operations such as **addition, subtraction, multiplication, and division** on two numbers entered by the user.

### CODE SNIPPET:

#### activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="20dp"
    android:background="@drawable/gradient_background"
    tools:context=".MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:orientation="vertical"
        android:background="@drawable/display_background"
        android:elevation="8dp"
        android:padding="24dp"
        android:layout_marginBottom="24dp">

        <TextView
            android:id="@+id/tvSecondaryDisplay"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="end"
            android:text=""
            android:textColor="#80FFFFFF"
            android:textSize="18sp"
            android:fontFamily="sans-serif-light"
            android:layout_marginBottom="8dp"
            android:minHeight="24dp" />

        <TextView
            android:id="@+id/tvDisplay"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="end|center_vertical"
            android:text="0"
            android:textColor="#FFFFFF"
            android:textSize="48sp"
```

```
    android:fontFamily="sans-serif-light"
    android:layout_gravity="bottom"
    android:maxLines="1"
    android:autoSizeTextType="uniform"
    android:autoSizeMinTextSize="24sp"
    android:autoSizeMaxTextSize="48sp" />

</LinearLayout>

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

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_marginBottom="12dp">

        <Button
            android:id="@+id/btnClear"
            style="@style/CalculatorButtonSecondary"
            android:layout_width="0dp"
            android:layout_weight="2"
            android:text="AC"
            android:textSize="20sp" />

        <Button
            android:id="@+id/btnBackspace"
            style="@style/CalculatorButtonSecondary"
            android:layout_width="0dp"
            android:layout_weight="1"
            android:text="⌫"
            android:textSize="20sp" />

        <Button
            android:id="@+id/btnDivide"
            style="@style/CalculatorButtonOperator"
            android:layout_width="0dp"
            android:layout_weight="1"
            android:text="÷" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_marginBottom="12dp">
```

```
<Button
    android:id="@+id	btn7"
    style="@style/CalculatorButtonNumber"
    android:layout_width="0dp"
    android:layout_weight="1"
    android:text="7" />

<Button
    android:id="@+id	btn8"
    style="@style/CalculatorButtonNumber"
    android:layout_width="0dp"
    android:layout_weight="1"
    android:text="8" />

<Button
    android:id="@+id	btn9"
    style="@style/CalculatorButtonNumber"
    android:layout_width="0dp"
    android:layout_weight="1"
    android:text="9" />

<Button
    android:id="@+id	btnMultiply"
    style="@style/CalculatorButtonOperator"
    android:layout_width="0dp"
    android:layout_weight="1"
    android:text="×" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginBottom="12dp">

<Button
    android:id="@+id	btn4"
    style="@style/CalculatorButtonNumber"
    android:layout_width="0dp"
    android:layout_weight="1"
    android:text="4" />

<Button
    android:id="@+id	btn5"
    style="@style/CalculatorButtonNumber"
    android:layout_width="0dp"
    android:layout_weight="1"
    android:text="5" />

<Button
    android:id="@+id	btn6"
    style="@style/CalculatorButtonNumber"
    android:layout_width="0dp"
```

```
    android:layout_weight="1"
    android:text="6" />

<Button
    android:id="@+id/btnSubtract"
    style="@style/CalculatorButtonOperator"
    android:layout_width="0dp"
    android:layout_weight="1"
    android:text="-" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginBottom="12dp">

    <Button
        android:id="@+id	btn1"
        style="@style/CalculatorButtonNumber"
        android:layout_width="0dp"
        android:layout_weight="1"
        android:text="1" />

    <Button
        android:id="@+id	btn2"
        style="@style/CalculatorButtonNumber"
        android:layout_width="0dp"
        android:layout_weight="1"
        android:text="2" />

    <Button
        android:id="@+id	btn3"
        style="@style/CalculatorButtonNumber"
        android:layout_width="0dp"
        android:layout_weight="1"
        android:text="3" />

    <Button
        android:id="@+id	btnAdd"
        style="@style/CalculatorButtonOperator"
        android:layout_width="0dp"
        android:layout_weight="1"
        android:text"+" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <Button
        android:id="@+id	btn0"
        style="@style/CalculatorButtonNumber"
        android:layout_width="0dp"
        android:layout_weight="1"
        android:text="0" />
</LinearLayout>
```

```
style="@style/CalculatorButtonNumber"
    android:layout_width="0dp"
    android:layout_weight="2"
    android:text="0" />

<Button
    android:id="@+id/btnDecimal"
    style="@style/CalculatorButtonNumber"
    android:layout_width="0dp"
    android:layout_weight="1"
    android:text"." />

<Button
    android:id="@+id/btnEquals"
    style="@style/CalculatorButtonEquals"
    android:layout_width="0dp"
    android:layout_weight="1"
    android:text="=" />
</LinearLayout>

</LinearLayout>
```

## MainActivity.java

```
package com.example.calculator;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;

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 implements
View.OnClickListener {

    private TextView tvDisplay;
    private TextView tvSecondaryDisplay;
```

```
private String currentNumber = "";  
private String operator = "";  
private String firstNumber = "";  
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.main),  
(v, insets) -> {
```

Insets systemBars =

```
    insets.getInsets(WindowInsetsCompat.Type.systemBars());  
    v.setPadding(systemBars.left, systemBars.top, systemBars.right,  
    systemBars.bottom);  
    return insets;  
});
```

```
initializeViews();  
setClickListeners();  
}
```

```
private void initializeViews() {  
    tvDisplay = findViewById(R.id.tvDisplay);  
    tvSecondaryDisplay = findViewById(R.id.tvSecondaryDisplay);  
}
```

```
private void setClickListeners() {  
    findViewById(R.id.btn0).setOnClickListener(this);  
    findViewById(R.id.btn1).setOnClickListener(this);  
    findViewById(R.id.btn2).setOnClickListener(this);  
    findViewById(R.id.btn3).setOnClickListener(this);  
    findViewById(R.id.btn4).setOnClickListener(this);  
    findViewById(R.id.btn5).setOnClickListener(this);  
    findViewById(R.id.btn6).setOnClickListener(this);  
    findViewById(R.id.btn7).setOnClickListener(this);  
    findViewById(R.id.btn8).setOnClickListener(this);  
    findViewById(R.id.btn9).setOnClickListener(this);
```

```
findViewById(R.id.btnAdd).setOnClickListener(this);
findViewById(R.id.btnSubtract).setOnClickListener(this);
findViewById(R.id.btnMultiply).setOnClickListener(this);
findViewById(R.id.btnDivide).setOnClickListener(this);
findViewById(R.id.btnEquals).setOnClickListener(this);

findViewById(R.id.btnClear).setOnClickListener(this);
findViewById(R.id.btnBackspace).setOnClickListener(this);
findViewById(R.id.btnDecimal).setOnClickListener(this);
```

```
}
```

```
@Override
```

```
public void onClick(View v) {
    Button button = (Button) v;
    String buttonText = button.getText().toString();

    int id = v.getId();

    if (id == R.id.btn0 || id == R.id.btn1 || id == R.id.btn2 || id == R.id.btn3 ||
        id == R.id.btn4 || id == R.id.btn5 || id == R.id.btn6 || id == R.id.btn7 ||
        id == R.id.btn8 || id == R.id.btn9) {
        handleNumberInput(buttonText);
    } else if (id == R.id.btnDecimal) {
        handleDecimalInput();
    } else if (id == R.id.btnAdd || id == R.id.btnSubtract || id == R.id.btnMultiply ||
    || id == R.id.btnDivide) {
        handleOperatorInput(buttonText);
    } else if (id == R.id.btnEquals) {
        handleEqualsInput();
    } else if (id == R.id.btnClear) {
        handleClearInput();
    } else if (id == R.id.btnBackspace) {
        handleBackspaceInput();
    }
}
```

```
private void handleNumberInput(String number) {
    if (isNewOperation) {
```

```
currentNumber = number;
isNewOperation = false;
} else {
    if (currentNumber.equals("0")) {
        currentNumber = number;
    } else {
        currentNumber += number;
    }
}
updateDisplay(currentNumber);
}

private void handleDecimalInput() {
    if (isNewOperation) {
        currentNumber = "0.";
        isNewOperation = false;
    } else if (!currentNumber.contains(".")) {
        if (currentNumber.isEmpty()) {
            currentNumber = "0.";
        } else {
            currentNumber += ".";
        }
    }
    updateDisplay(currentNumber);
}

private void handleOperatorInput(String op) {
    if (!currentNumber.isEmpty()) {
        if (!firstNumber.isEmpty() && !operator.isEmpty()) {
            handleEqualsInput();
            firstNumber = currentNumber;
        } else {
            firstNumber = currentNumber;
        }
    }

    String displayOp = op;
    switch (op) {
        case "×":
            operator = "*";
            break;
```

```
        case "/":
            operator = "/";
            break;
        case "-":
            operator = "-";
            displayOp = "-";
            break;
        default:
            operator = op;
            break;
    }

    updateSecondaryDisplay(firstNumber + " " + displayOp);
    currentNumber = "";
    isNewOperation = true;
}

}

private void handleEqualsInput() {
    if (!firstNumber.isEmpty() && !currentNumber.isEmpty() &&
!operator.isEmpty()) {
        try {
            double num1 = Double.parseDouble(firstNumber);
            double num2 = Double.parseDouble(currentNumber);
            double result = 0;

            switch (operator) {
                case "+":
                    result = num1 + num2;
                    break;
                case "-":
                    result = num1 - num2;
                    break;
                case "*":
                    result = num1 * num2;
                    break;
                case "/":
                    if (num2 != 0) {
```

```
        result = num1 / num2;
    } else {
        Toast.makeText(this, "Cannot divide by zero",
Toast.LENGTH_SHORT).show();
        return;
    }
    break;
}

String resultString;
if(result == (long) result) {
    resultString = String.valueOf((long) result);
} else {
    resultString = String.valueOf(result);
}

currentNumber = resultString;
updateDisplay(currentNumber);

firstNumber = "";
operator = "";
isNewOperation = true;
updateSecondaryDisplay("");


} catch (NumberFormatException e) {
    Toast.makeText(this, "Invalid calculation",
Toast.LENGTH_SHORT).show();
    handleClearInput();
}
}

private void handleClearInput() {
    currentNumber = "";
    firstNumber = "";
    operator = "";
    isNewOperation = true;
    updateDisplay("0");
    updateSecondaryDisplay("");
}
```

```
private void handleBackspaceInput() {
    if (!currentNumber.isEmpty() && !isNewOperation) {
        currentNumber = currentNumber.substring(0, currentNumber.length() - 1);
        if (currentNumber.isEmpty()) {
            updateDisplay("0");
            isNewOperation = true;
        } else {
            updateDisplay(currentNumber);
        }
    }
}

private void updateDisplay(String value) {
    if (value.isEmpty()) {
        tvDisplay.setText("0");
    } else {
        tvDisplay.setText(value);
    }
}

private void updateSecondaryDisplay(String value) {
    tvSecondaryDisplay.setText(value);
}
```

## res/values/themes.xml

```
<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Base.Theme.Calculator"
        parent="Theme.Material3.DayNight.Dialog.Alert">
        <item name="colorPrimary">#1E1E1E</item>
        <item name="colorPrimaryVariant">#000000</item>
        <item name="colorOnPrimary">#FFFFFF</item>
        <item name="android:statusBarColor">@android:color/transparent</item>
        <item
            name="android:navigationBarColor">@android:color/transparent</item>
            <item name="android:windowLightStatusBar">false</item>
    </style>
```

```
<style name="Theme.Calculator" parent="Base.Theme.Calculator" />

<style name="CalculatorButtonBase"
parent="Widget.Material3.Button.UnelevatedButton">
    <item name="android:layout_height">72dp</item>
    <item name="android:layout_margin">6dp</item>
    <item name="android:textSize">28sp</item>
    <item name="android:fontFamily">sans-serif-light</item>
    <item name="cornerRadius">20dp</item>
    <item name="android:elevation">0dp</item>
    <item name="android:stateListAnimator">@null</item>
    <item name="rippleColor">#40FFFFFF</item>
</style>

<style name="CalculatorButtonNumber" parent="CalculatorButtonBase">
    <item name="android:textColor">#FFFFFF</item>
    <item name="backgroundTint">#2D2D2D</item>
    <item name="android:textSize">32sp</item>
</style>

<style name="CalculatorButtonOperator" parent="CalculatorButtonBase">
    <item name="android:textColor">#FFFFFF</item>
    <item name="backgroundTint">#FF9500</item>
    <item name="android:textSize">32sp</item>
    <item name="android:fontFamily">sans-serif</item>
</style>

<style name="CalculatorButtonSecondary" parent="CalculatorButtonBase">
    <item name="android:textColor">#1E1E1E</item>
    <item name="backgroundTint">#A6A6A6</item>
    <item name="android:textSize">24sp</item>
    <item name="android:fontFamily">sans-serif-medium</item>
</style>

<style name="CalculatorButtonEquals" parent="CalculatorButtonBase">
    <item name="android:textColor">#FFFFFF</item>
    <item name="backgroundTint">#FF9500</item>
    <item name="android:textSize">36sp</item>
    <item name="android:fontFamily">sans-serif-light</item>
</style>
</resources>
```

### **res/drawable/button\_number\_selector.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:state_pressed="true">
        <shape android:shape="rectangle">
            <solid android:color="#404040" />
            <corners android:radius="20dp" />
        </shape>
    </item>
    <item>
        <shape android:shape="rectangle">
            <solid android:color="#2D2D2D" />
            <corners android:radius="20dp" />
        </shape>
    </item>
</selector>
```

### **res/drawable/button\_operator\_selector.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:state_pressed="true">
        <shape android:shape="rectangle">
            <solid android:color="#FFB84D" />
            <corners android:radius="20dp" />
        </shape>
    </item>
    <item>
        <shape android:shape="rectangle">
            <solid android:color="#FF9500" />
            <corners android:radius="20dp" />
        </shape>
    </item>
</selector>
```

### **res/drawable/display\_background.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <gradient
        android:angle="90"
        android:startColor="#3A3A3A"
        android:endColor="#2A2A2A"
        android:type="linear" />
```

```
<corners android:radius="24dp" />
<stroke
    android:width="1dp"
    android:color="#4A4A4A" />
</shape>
```

### **res/drawable/gradient\_background.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <gradient
        android:angle="135"
        android:startColor="#1E1E1E"
        android:centerColor="#2D2D2D"
        android:endColor="#1A1A1A"
        android:type="linear" />
</shape>
```

### **OUTPUT:**