state handling

How data is preserved when the **screen rotates** or the **activity restarts** (which triggers the Activity lifecycle again).

When the user enters two numbers and presses **Add**, the result should still be visible even if:

- The **screen is rotated**, or
- The **app temporarily goes to background and comes back**.

Step 1: Updated activity_main.xml

No major change — just ensure IDs are correct:

```
<?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:id="@+id/tvTitle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Simple Calculator"
    android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginTop="40dp"/>

  <EditText
    android:id="@+id/etNum1"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:hint="Enter number 1"
    android:inputType="number"
    app:layout_constraintTop_toBottomOf="@id/tvTitle"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_margin="20dp"/>

  <EditText
    android:id="@+id/etNum2"
    android:layout_width="0dp"
```

```xml
    android:layout_height="wrap_content"
    android:hint="Enter number 2"
    android:inputType="number"
    app:layout_constraintTop_toBottomOf="@id/etNum1"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_margin="20dp"/>

  <Button
    android:id="@+id/btnAdd"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Add Numbers"
    app:layout_constraintTop_toBottomOf="@id/etNum2"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"/>

  <TextView
    android:id="@+id/tvResult"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Result will appear here"
    android:textSize="18sp"
    app:layout_constraintTop_toBottomOf="@id/btnAdd"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginTop="20dp"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Step 2: Updated MainActivity.kt

Kotlin code with state handling using onSaveInstanceState() and onRestoreInstanceState().

```kotlin
package com.example.lifecycledemo

import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

    private lateinit var etNum1: EditText
    private lateinit var etNum2: EditText
    private lateinit var tvResult: TextView
    private lateinit var btnAdd: Button

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

        etNum1 = findViewById(R.id.etNum1)
        etNum2 = findViewById(R.id.etNum2)
        tvResult = findViewById(R.id.tvResult)
        btnAdd = findViewById(R.id.btnAdd)

        btnAdd.setOnClickListener {
            val num1 = etNum1.text.toString().toDoubleOrNull() ?: 0.0
            val num2 = etNum2.text.toString().toDoubleOrNull() ?: 0.0
            val sum = num1 + num2
            tvResult.text = "Result: $sum"
        }

        // Restore saved state if available
        if (savedInstanceState != null) {
            val savedResult = savedInstanceState.getString("RESULT_TEXT")
            tvResult.text = savedResult
        }
    }

    // Called before the activity is destroyed (e.g., rotation)
    override fun onSaveInstanceState(outState: Bundle) {
        super.onSaveInstanceState(outState)
        outState.putString("RESULT_TEXT", tvResult.text.toString())
    }

    // Optional: alternative way to restore after rotation
    override fun onRestoreInstanceState(savedInstanceState: Bundle) {
        super.onRestoreInstanceState(savedInstanceState)
        val savedResult = savedInstanceState.getString("RESULT_TEXT")
        tvResult.text = savedResult
    }
}
```

Explanation

Lifecycle Method	**Purpose**
onSaveInstanceState()	Called *before* the Activity is destroyed (e.g., on rotation). We store key-value pairs here.
onRestoreInstanceState()	Called *after* the Activity is recreated. We retrieve saved values and restore them.
savedInstanceState (in onCreate)	When the Activity starts again, this Bundle may contain previously saved data.

Run the app

1. Run the app.
2. Enter two numbers and press **Add** → result appears.
3. Rotate the device (Ctrl + F11 in Emulator).

Note: The result text stays visible — no reset!