**//Design**



**// Code**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

/\*\*

\* Represents a simple calculator application with a Swing GUI.

\*/

public class SimpleCalculator extends JFrame {

private JTextField textField1, textField2, resultField;

/\*\*

\* Constructs a SimpleCalculator GUI.

\*/

public SimpleCalculator() {

// Set up the JFrame

setTitle("Simple Calculator");

setSize(400, 250); // Increased height to accommodate the button

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLocationRelativeTo(null);

// Create components

textField1 = new JTextField(10);

textField2 = new JTextField(10);

JButton calculateButton = new JButton("Calculate");

resultField = new JTextField(10);

resultField.setEditable(false); // Make it non-editable

resultField.setBackground(Color.WHITE); // Set background color to white

// Create a JPanel for input components with GridLayout

JPanel inputPanel = new JPanel(new GridLayout(2, 2, 5, 1)); // 2 rows, 2 columns, gaps

inputPanel.add(new JLabel("First Value"));

inputPanel.add(textField1);

inputPanel.add(new JLabel("Second Value"));

inputPanel.add(textField2);

// Create a JPanel for the result with GridLayout

JPanel resultPanel = new JPanel(new GridLayout(1, 2, 5, 5)); // 1 row, 2 columns, gaps

resultPanel.add(new JLabel("Sum is"));

resultPanel.add(resultField);

// Create a JPanel for the button

JPanel buttonPanel = new JPanel();

buttonPanel.add(calculateButton);

// Add panels to the frame

add(inputPanel, BorderLayout.PAGE\_START); // PAGE\_START to make it above the resultPanel

add(resultPanel, BorderLayout.CENTER);

add(buttonPanel, BorderLayout.PAGE\_END); // PAGE\_END corresponds to bottom center

// Add action listener to the Calculate button

calculateButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

calculateSum();

}

});

pack();

setLocationRelativeTo(null);

}

/\*\*

\* Calculates the sum of the values in textField1 and textField2 and updates resultField.

\*/

private void calculateSum() {

try {

double value1 = Double.parseDouble(textField1.getText());

double value2 = Double.parseDouble(textField2.getText());

double sum = value1 + value2;

resultField.setText(String.valueOf(sum));

} catch (NumberFormatException ex) {

// Handle the case where input is not a valid double

JOptionPane.showMessageDialog(this, "Please enter valid numbers for calculation.",

"Invalid Input", JOptionPane.ERROR\_MESSAGE);

}

}

/\*\*

\* The entry point of the SimpleCalculator application.

\*

\* @param args Command-line arguments (not used in this application).

\*/

public static void main(String[] args) {

SwingUtilities.invokeLater(new Runnable() {

@Override

public void run() {

new SimpleCalculator().setVisible(true);

}

});

}

}

**//Output**

