```java
// SimpleCalculator
//IntelliJ IDEA 2023.1

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class SimpleCalculator extends JFrame implements ActionListener {

    private JTextField firstNumberField;
    private JTextField secondNumberField;
    private JTextField answerField;
    private char operator;

    public SimpleCalculator() {
        setTitle("Simple Calculator");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());

        // Panels
        JPanel topPanel = new JPanel();
        topPanel.setLayout(new GridLayout(3, 2, 10, 10));

        JPanel centerPanel = new JPanel();
        centerPanel.setLayout(new GridLayout(4, 4, 10, 10));

        // Labels and Text Fields
        JLabel firstNumberLabel = new JLabel("Enter first number:");
```

```java
firstNumberField = new JTextField(10);

JLabel secondNumberLabel = new JLabel("Enter second number:");
secondNumberField = new JTextField(10);

JLabel answerLabel = new JLabel("Answer:");
answerField = new JTextField(10);
answerField.setEditable(false);

// Buttons
JButton addButton = new JButton("+");
JButton subtractButton = new JButton("-");
JButton multiplyButton = new JButton("*");
JButton divideButton = new JButton("/");
JButton clearButton = new JButton("Clear");

addButton.setFont(new Font("Arial", Font.PLAIN, 20));
subtractButton.setFont(new Font("Arial", Font.PLAIN, 20));
multiplyButton.setFont(new Font("Arial", Font.PLAIN, 20));
divideButton.setFont(new Font("Arial", Font.PLAIN, 20));
clearButton.setFont(new Font("Arial", Font.PLAIN, 20));

addButton.addActionListener(this);
subtractButton.addActionListener(this);
multiplyButton.addActionListener(this);
divideButton.addActionListener(this);
clearButton.addActionListener(this);

centerPanel.add(addButton);
```

```java
        centerPanel.add(subtractButton);

        centerPanel.add(multiplyButton);

        centerPanel.add(divideButton);

        centerPanel.add(clearButton);


        // Add components to panels

        topPanel.add(firstNumberLabel);

        topPanel.add(firstNumberField);

        topPanel.add(secondNumberLabel);

        topPanel.add(secondNumberField);

        topPanel.add(answerLabel);

        topPanel.add(answerField);


        // Add panels to frame

        add(topPanel, BorderLayout.NORTH);

        add(centerPanel, BorderLayout.CENTER);


        pack();

        setLocationRelativeTo(null); // Center the frame on the screen

    }


    @Override

    public void actionPerformed(ActionEvent e) {

        String actionCommand = e.getActionCommand();


        if (actionCommand.equals("+") || actionCommand.equals("-") || actionCommand.equals("*") ||
actionCommand.equals("/")) {

            operator = actionCommand.charAt(0);

            calculate();
```

```java
    } else if (actionCommand.equals("Clear")) {

        clear();

    }

}


private void calculate() {

    try {

        double firstNumber = Double.parseDouble(firstNumberField.getText());

        double secondNumber = Double.parseDouble(secondNumberField.getText());


        double result = 0;


        switch (operator) {

            case '+':

                result = firstNumber + secondNumber;

                break;

            case '-':

                result = firstNumber - secondNumber;

                break;

            case '*':

                result = firstNumber * secondNumber;

                break;

            case '/':

                result = firstNumber / secondNumber;

                break;

        }


        answerField.setText(String.valueOf(result));

    } catch (NumberFormatException e) {
```

```java
                    answerField.setText("Error: Invalid input");
                }
            }


            private void clear() {
                firstNumberField.setText("");
                secondNumberField.setText("");
                answerField.setText("");
                operator = ' ';
            }


            public static void main(String[] args) {
                SwingUtilities.invokeLater(() -> {
                    SimpleCalculator calculator = new SimpleCalculator();
                    calculator.setVisible(true);
                });
            }
        }
```