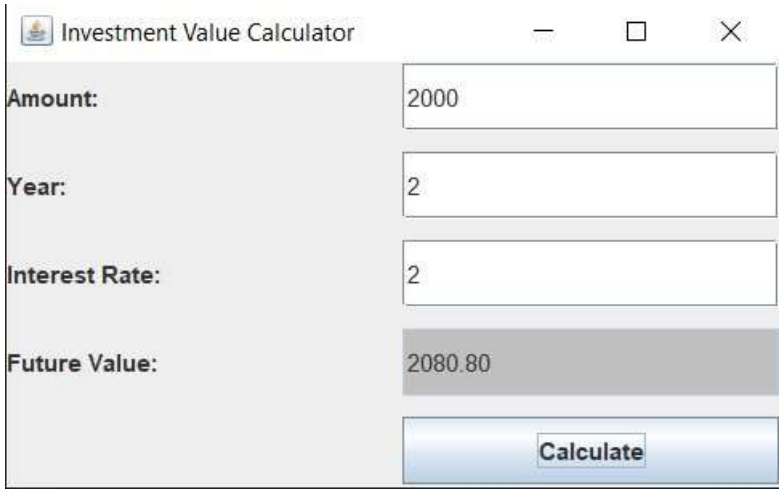
# Practical 12:- GUI

## 1. Write a programme to implement an investement value calculator using the data inputed by user. textFields to be included are amount, year, interest rate and future value. The field “future value” (shown in gray) must not be altered by user.

```
import java.awt.*;
import javax.swing.*;

public class InvestmentCalculator extends JFrame {
   private JTextField amountField, yearField, rateField, futureValueField;

   public InvestmentCalculator() {
setTitle("Investment Value Calculator");
setLayout(new GridLayout(5, 2, 10, 10));
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setSize(400, 250);
      setLocationRelativeTo(null); // Center the frame

      // Labels
      add(new JLabel("Amount:"));
amountField = new JTextField();
add(amountField);

      add(new JLabel("Year:"));
yearField = new JTextField();
add(yearField);

      add(new JLabel("Interest Rate:"));
rateField = new JTextField();
      add(rateField);

      add(new JLabel("Future Value:"));
futureValueField = new JTextField();
      futureValueField.setEditable(false); // Make it non-editable
futureValueField.setBackground(Color.LIGHT_GRAY);       add(futureValueField);
```

```
        JButton calculateButton = new JButton("Calculate");
add(new JLabel()); // Empty label for spacing
add(calculateButton);

        // Action listener
        calculateButton.addActionListener(e -> calculateFutureValue());

        setVisible(true);
    }

    private void calculateFutureValue() {
try {
            double amount = Double.parseDouble(amountField.getText());
int years = Integer.parseInt(yearField.getText());          double rate =
Double.parseDouble(rateField.getText());

            double futureValue = amount * Math.pow(1 + rate / 100, years);
futureValueField.setText(String.format("%.2f", futureValue));
        } catch (NumberFormatException ex) {
            JOptionPane.showMessageDialog(this, "Please enter valid numeric values.", "Input
Error", JOptionPane.ERROR_MESSAGE);
        }
    }

    public static void main(String[] args) {
        new InvestmentCalculator();
    }
}
```

## Output :-



## 2. Write a program which fill the rectangle with the selected color when button pressed.

```
import javax.swing.*; import
java.awt.*;
import java.awt.event.*;

public class ColorChanger extends JFrame {
private JPanel colorPanel;    private JLabel
statusLabel;

   public ColorChanger() {
setTitle("I am a JFrame");
setSize(400, 220);
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setLayout(new GridLayout(1, 2)); // Two side-by-side panels

      // Left Panel (buttons + label)
      JPanel buttonPanel = new JPanel(new GridLayout(4, 1, 10, 10));
buttonPanel.setBackground(Color.GREEN);

      JButton redBtn = new JButton("Red");
      JButton blueBtn = new JButton("Blue");
      JButton greenBtn = new JButton("Green");

      statusLabel = new JLabel("Select a color", SwingConstants.CENTER);
statusLabel.setFont(new Font("Arial", Font.BOLD, 12));

      buttonPanel.add(redBtn);
buttonPanel.add(blueBtn);
buttonPanel.add(greenBtn);
      buttonPanel.add(statusLabel);

      // Right Panel (color display area)
colorPanel = new JPanel();
      colorPanel.setBackground(Color.BLUE); // Default color
      colorPanel.setBorder(BorderFactory.createLineBorder(Color.BLACK, 5)); // Add border

      // Button actions
      redBtn.addActionListener(e -> {
colorPanel.setBackground(Color.RED);          statusLabel.setText("You
selected: Red");
      });
```

```
        blueBtn.addActionListener(e -> {
colorPanel.setBackground(Color.BLUE);
            statusLabel.setText("You selected: Blue");
        });

        greenBtn.addActionListener(e -> {
colorPanel.setBackground(Color.GREEN);
statusLabel.setText("You selected: Green");
        });

        // Add both panels to frame
add(buttonPanel);
        add(colorPanel);

        setVisible(true);
    }

    public static void main(String[] args) {
new ColorChanger();
    }
}
```

## Output :-