a)Construct a simple calculator using the JAVA Swings with minimum functionality.

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

public class SimpleCalculator extends JFrame {
    private JTextField display;
    private JButton[] numberButtons;
    private JButton addButton, subtractButton, multiplyButton, divideButton, equalsButton,
clearButton;

    private double firstNumber, secondNumber, result;
    private String operator;

    public SimpleCalculator() {
        setTitle("Simple Calculator");
        setSize(300, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        createComponents();
        setLayout();
    }

    private void createComponents() {
        display = new JTextField();
        display.setEditable(false);

        numberButtons = new JButton[10];
        for (int i = 0; i < 10; i++) {
            numberButtons[i] = new JButton(String.valueOf(i));
            numberButtons[i].addActionListener(new NumberButtonActionListener());
        }

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

        addButton.addActionListener(new OperationButtonActionListener());
        subtractButton.addActionListener(new OperationButtonActionListener());
```

```
    multiplyButton.addActionListener(new OperationButtonActionListener());
    divideButton.addActionListener(new OperationButtonActionListener());
    equalsButton.addActionListener(new EqualsButtonActionListener());
    clearButton.addActionListener(new ClearButtonActionListener());
}

private void setLayout() {
    JPanel panel = new JPanel();
    panel.setLayout(new GridLayout(4, 4));

    // Add number buttons to the panel
    for (int i = 1; i <= 9; i++) {
        panel.add(numberButtons[i]);
    }
    panel.add(addButton);
    panel.add(numberButtons[0]);
    panel.add(subtractButton);
    panel.add(multiplyButton);
    panel.add(divideButton);
    panel.add(equalsButton);
    panel.add(clearButton);

    setLayout(new BorderLayout());
    add(display, BorderLayout.NORTH);
    add(panel, BorderLayout.CENTER);
}

private class NumberButtonActionListener implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent e) {
        JButton buttonClicked = (JButton) e.getSource();
        display.setText(display.getText() + buttonClicked.getText());
    }
}

private class OperationButtonActionListener implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent e) {
        JButton buttonClicked = (JButton) e.getSource();
        firstNumber = Double.parseDouble(display.getText());
        operator = buttonClicked.getText();
        display.setText("");
    }
}
```

```
private class EqualsButtonActionListener implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent e) {
        secondNumber = Double.parseDouble(display.getText());

        switch (operator) {
            case "+":
                result = firstNumber + secondNumber;
                break;
            case "-":
                result = firstNumber - secondNumber;
                break;
            case "*":
                result = firstNumber * secondNumber;
                break;
            case "/":
                if (secondNumber != 0) {
                    result = firstNumber / secondNumber;
                } else {
                    display.setText("Error");
                    return;
                }
                break;
        }

        display.setText(String.valueOf(result));
    }
}

private class ClearButtonActionListener implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent e) {
        display.setText("");
        firstNumber = 0;
        secondNumber = 0;
        result = 0;
        operator = null;
    }
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        @Override
```

```
            public void run() {
                new SimpleCalculator().setVisible(true);
            }
        });
    }
}
```

B)Construct a GUI using JAVA Swings to accept details of a record of a given table and submit it to the database using JDBC technology on the click of a button.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class RecordForm extends JFrame {

    private JTextField nameField;
    private JTextField ageField;
    private JButton submitButton;

    public RecordForm() {
        setTitle("Record Form");
        setSize(300, 200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        createComponents();
        setLayout();
    }

    private void createComponents() {
        nameField = new JTextField();
        ageField = new JTextField();
        submitButton = new JButton("Submit");
        submitButton.addActionListener(new SubmitButtonActionListener());
    }

    private void setLayout() {
        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(3, 2));
```

```
        panel.add(new JLabel("Name:"));
        panel.add(nameField);
        panel.add(new JLabel("Age:"));
        panel.add(ageField);
        panel.add(new JLabel()); // Placeholder
        panel.add(submitButton);

        setLayout(new BorderLayout());
        add(panel, BorderLayout.CENTER);
    }

    private class SubmitButtonActionListener implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent e) {
            String name = nameField.getText();
            String age = ageField.getText();

            // Validate input
            if (name.isEmpty() || age.isEmpty()) {
                JOptionPane.showMessageDialog(RecordForm.this, "Please enter valid details",
"Error", JOptionPane.ERROR_MESSAGE);
                return;
            }

            // Database connection parameters
            String url = "jdbc:hsqldb:mem:testdb";
            String user = "SA";
            String password = "";

            try (Connection connection = DriverManager.getConnection(url, user, password)) {
                // Create table if not exists
                String createTableSQL = "CREATE TABLE IF NOT EXISTS records (id INTEGER
IDENTITY, name VARCHAR(255), age INTEGER)";
                try (PreparedStatement createTableStatement =
connection.prepareStatement(createTableSQL)) {
                    createTableStatement.execute();
                }

                // Insert record
                String insertRecordSQL = "INSERT INTO records (name, age) VALUES (?, ?)";
                try (PreparedStatement insertStatement =
connection.prepareStatement(insertRecordSQL)) {
                    insertStatement.setString(1, name);
```

```
                insertStatement.setInt(2, Integer.parseInt(age));
                insertStatement.executeUpdate();
            }

            JOptionPane.showMessageDialog(RecordForm.this, "Record submitted successfully",
"Success", JOptionPane.INFORMATION_MESSAGE);

            // Clear fields after submission
            nameField.setText("");
            ageField.setText("");

        } catch (SQLException ex) {
            ex.printStackTrace();
            JOptionPane.showMessageDialog(RecordForm.this, "Error submitting record",
"Error", JOptionPane.ERROR_MESSAGE);
        }
    }
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        @Override
        public void run() {
            new RecordForm().setVisible(true);
        }
    });
}
}
```