1. Write a Java Swing program to create a GUI arithmetic calculator (Labels, Text boxes, Buttons and also use an ActionListener interface)

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
public class Calculator extends JFrame implements ActionListener {
    private JTextField num1Field;
    private JTextField num2Field;
    private JTextField resultField;
    private JButton addButton;
    private JButton subtractButton;
    private JButton multiplyButton;
    private JButton divideButton;

public Calculator() {
    setTitle("Simple Calculator");
    setSize(400, 300);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setLayout(new GridLayout(5, 2));

    num1Field = new JTextField();
    num2Field = new JTextField();
    resultField = new JTextField();
    resultField.setEditable(false);

    addButton = new JButton("Add");
    subtractButton = new JButton("Subtract");
    multiplyButton = new JButton("Multiply");
    divideButton = new JButton("Divide");

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

    add(new JLabel("Number 1:"));
```

```
        add(num1Field);
        add(new JLabel("Number 2:"));
        add(num2Field);
        add(new JLabel("Result:"));
        add(resultField);
        add(addButton);
        add(subtractButton);
        add(multiplyButton);
        add(divideButton);
    }
@Override
    public void actionPerformed(ActionEvent e) {
        double num1 = Double.parseDouble(num1Field.getText());
        double num2 = Double.parseDouble(num2Field.getText());
        double result = 0;
        if (e.getSource() == addButton) {
            result = num1 + num2;
        } else if (e.getSource() == subtractButton) {
            result = num1 - num2;
        } else if (e.getSource() == multiplyButton) {
            result = num1 * num2;
        } else if (e.getSource() == divideButton) {
            if (num2 != 0) {
                result = num1 / num2;
            } else {
                JOptionPane.showMessageDialog(this, "Cannot divide by zero", "Error",
JOptionPane.ERROR_MESSAGE);
                return;
            }
        } resultField.setText(String.valueOf(result));
    }
    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            Calculator calculator = new Calculator();
            calculator.setVisible(true);
        });
    }
}
```

2. Write a Java Swing program to create a frame containing three buttons (Hello, Welcome, Bye). When clicked on button Hello, Welcome or Bye the message "Hello Friends", “Welcome to Ranchi” or “Bye Friends” gets displayed in label control.

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

public class MessageDisplayApp extends JFrame implements
   ActionListener {
        private JLabel messageLabel;
        private JButton helloButton;
        private JButton welcomeButton;
        private JButton byeButton;

public MessageDisplayApp() {
      setTitle("Message Display App");
      setSize(300, 200);
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      setLayout(new FlowLayout());

      messageLabel = new JLabel("Click a button to see a message");
      helloButton = new JButton("Hello");
      welcomeButton = new JButton("Welcome");
      byeButton = new JButton("Bye");

      helloButton.addActionListener(this);
      welcomeButton.addActionListener(this);
      byeButton.addActionListener(this);

      add(messageLabel);
      add(helloButton);
      add(welcomeButton);
      add(byeButton);
   }
```

```
 @Override
   public void actionPerformed(ActionEvent e) {
      if (e.getSource() == helloButton) {
         messageLabel.setText("Hello Friends");
      } else if (e.getSource() == welcomeButton) {
         messageLabel.setText("Welcome to Ranchi");
      } else if (e.getSource() == byeButton) {
         messageLabel.setText("Bye Friends");
      }
   }
   public static void main(String[] args) {
      SwingUtilities.invokeLater(() -> {
         MessageDisplayApp app = new MessageDisplayApp();
         app.setVisible(true);
      });
   }
}
```

3. Write a Java Swing program to create three radio buttons (red, green and blue). When any of them is selected, to change the background color of TextField with suitable message

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

public class ColorChangeApp extends JFrame implements ActionListener {
     private JTextField textField;
     private JRadioButton redButton;
     private JRadioButton greenButton;
     private JRadioButton blueButton;
     private ButtonGroup colorGroup;

public ColorChangeApp() {
       setTitle("Color Change App");
       setSize(300, 200);
```

```
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setLayout(new FlowLayout());

    textField = new JTextField("Select a color", 20);
    textField.setEditable(false);

    redButton = new JRadioButton("Red");
    greenButton = new JRadioButton("Green");
    blueButton = new JRadioButton("Blue");

    colorGroup = new ButtonGroup();
    colorGroup.add(redButton);
    colorGroup.add(greenButton);
    colorGroup.add(blueButton);

    redButton.addActionListener(this);
    greenButton.addActionListener(this);
    blueButton.addActionListener(this);

    add(textField);
    add(redButton);
    add(greenButton);
    add(blueButton);
}
@Override
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == redButton) {
        textField.setBackground(Color.RED);
        textField.setText("You selected Red");
    } else if (e.getSource() == greenButton) {
        textField.setBackground(Color.GREEN);
        textField.setText("You selected Green");
    } else if (e.getSource() == blueButton) {
        textField.setBackground(Color.BLUE);
        textField.setText("You selected Blue");
    }
}
```

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

4. Write a Java Swing program to create a combo box which includes list of subjects. When select a subject from the combo box then that subject will display in the Label with increases the font size and color.

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

public class SubjectSelectorApp extends JFrame implements ActionListener {
   private JComboBox<String> subjectComboBox;
   private JLabel subjectLabel;

public SubjectSelectorApp() {
      setTitle("Subject Selector");
      setSize(300, 200);
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      setLayout(new FlowLayout());

      String[] subjects = {"Mathematics", "Physics", "Chemistry", "Biology",
      "Computer Science"};
      subjectComboBox = new JComboBox<>(subjects);
      subjectLabel = new JLabel("Select a subject from the combo box");
      subjectComboBox.addActionListener(this);

      add(subjectComboBox);
      add(subjectLabel);
   }
```

```
 @Override
   public void actionPerformed(ActionEvent e) {
      String selectedSubject = (String) subjectComboBox.getSelectedItem();

      subjectLabel.setText(selectedSubject);
      subjectLabel.setFont(new Font("Arial", Font.BOLD, 24));
      subjectLabel.setForeground(Color.BLUE); // Change text color
   }
public static void main(String[] args) {
      SwingUtilities.invokeLater(() -> {
         SubjectSelectorApp app = new SubjectSelectorApp();
         app.setVisible(true);
      });
   }
}
```

5. Write a Java Swing program to create three radio buttons. When any of them is selected, an appropriate message is displayed in message box.

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

public class RadioButtonMessageApp extends JFrame implements
ActionListener {
   private JRadioButton option1;
   private JRadioButton option2;
   private JRadioButton option3;
   private ButtonGroup buttonGroup;

   public RadioButtonMessageApp() {
      setTitle("Radio Button Message App");
      setSize(300, 200);
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      setLayout(new FlowLayout());
```

```
        option1 = new JRadioButton("Option 1");
        option2 = new JRadioButton("Option 2");
        option3 = new JRadioButton("Option 3");

        buttonGroup = new ButtonGroup();
        buttonGroup.add(option1);
        buttonGroup.add(option2);
        buttonGroup.add(option3);

        option1.addActionListener(this);
        option2.addActionListener(this);
        option3.addActionListener(this);

        add(option1);
        add(option2);
        add(option3);
    }
@Override
      public void actionPerformed(ActionEvent e) {
           if (e.getSource() == option1) {
           JOptionPane.showMessageDialog(this, "You selected Option 1",
"Message", JOptionPane.INFORMATION_MESSAGE);
        } else if (e.getSource() == option2) {
           JOptionPane.showMessageDialog(this, "You selected Option 2",
"Message", JOptionPane.INFORMATION_MESSAGE);
        } else if (e.getSource() == option3) {
           JOptionPane.showMessageDialog(this, "You selected Option 3",
"Message", JOptionPane.INFORMATION_MESSAGE);
        }
    }
public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
           RadioButtonMessageApp app = new RadioButtonMessageApp();
           app.setVisible(true);
        });
    }
}
```

6. Write a Java program using Swing to create a list box to add some state names in the list box.

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

public class StateListApp extends JFrame {
   private JList<String> stateList;
   private DefaultListModel<String> listModel;
   private JTextField stateTextField;
   private JButton addButton;

public StateListApp() {
       setTitle("State List App");
       setSize(300, 300);
       setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
       setLayout(new FlowLayout());

      listModel = new DefaultListModel<>();
      stateList = new JList<>(listModel);
      stateList.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
       JScrollPane listScrollPane = new JScrollPane(stateList);
       listScrollPane.setPreferredSize(new Dimension(250, 150));

       stateTextField = new JTextField(15);
       addButton = new JButton("Add State");
       addButton.addActionListener(new ActionListener() {
          @Override
          public void actionPerformed(ActionEvent e) {
             String stateName = stateTextField.getText().trim();
             if (!stateName.isEmpty()) {
                listModel.addElement(stateName);
                stateTextField.setText("");
             } else {
```

```
                JOptionPane.showMessageDialog(StateListApp.this, "Please enter a
state name.", "Error", JOptionPane.ERROR_MESSAGE);
            }
        }
    });
    add(listScrollPane);
    add(stateTextField);
    add(addButton);
  }
public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
        StateListApp app = new StateListApp();
        app.setVisible(true);
    });
  }
}
```

7. Write a Java Swing program using multiple check boxes to develop Food Order System for a Restaurant.

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

public class FoodOrderSystem extends JFrame {
  private JCheckBox pizzaCheckBox;
  private JCheckBox burgerCheckBox;
  private JCheckBox pastaCheckBox;
  private JCheckBox saladCheckBox;
  private JCheckBox dessertCheckBox;
  private JButton orderButton;
  private JTextArea orderSummary;

  private final double PIZZA_PRICE = 8.99;
  private final double BURGER_PRICE = 5.99;
  private final double PASTA_PRICE = 7.49;
  private final double SALAD_PRICE = 4.99;
  private final double DESSERT_PRICE = 3.49;
```

```
 public FoodOrderSystem() {
      setTitle("Restaurant Food Order System");
      setSize(300, 300);
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      setLayout(new FlowLayout());

      pizzaCheckBox = new JCheckBox("Pizza - $8.99");
      burgerCheckBox = new JCheckBox("Burger - $5.99");
      pastaCheckBox = new JCheckBox("Pasta - $7.49");
      saladCheckBox = new JCheckBox("Salad - $4.99");
      dessertCheckBox = new JCheckBox("Dessert - $3.49");

      orderButton = new JButton("Place Order");
      orderSummary = new JTextArea(10, 25);
      orderSummary.setEditable(false);
      JScrollPane scrollPane = new JScrollPane(orderSummary);
      orderButton.addActionListener(new ActionListener()
{
@Override
         public void actionPerformed(ActionEvent e) {
            double totalCost = 0;
            StringBuilder summary = new StringBuilder("Order Summary:\n");
            if (pizzaCheckBox.isSelected()) {
               summary.append("Pizza\n");
               totalCost += PIZZA_PRICE;
            }
            if (burgerCheckBox.isSelected()) {
               summary.append("Burger\n");
               totalCost += BURGER_PRICE;
            }
            if (pastaCheckBox.isSelected()) {
               summary.append("Pasta\n");
               totalCost += PASTA_PRICE;
            }
            if (saladCheckBox.isSelected()) {
               summary.append("Salad\n");
               totalCost += SALAD_PRICE;
            }
```

```
            if (dessertCheckBox.isSelected()) {
                summary.append("Dessert\n");
                totalCost += DESSERT_PRICE;
            }
            summary.append("Total Cost: $").append(String.format("%.2f",
totalCost));
            orderSummary.setText(summary.toString());
        }
    });
    add(pizzaCheckBox);
    add(burgerCheckBox);
    add(pastaCheckBox);
    add(saladCheckBox);
    add(dessertCheckBox);
    add(orderButton);
    add(scrollPane);
  }
  public static void main(String[] args) {
    SwingUtilities.invokeLater(() -> {
      FoodOrderSystem app = new FoodOrderSystem();
      app.setVisible(true);
    });
  }
}
```

8. Write a Java Swing program with using JTree class give a suitable example.

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

public class JTreeExample extends JFrame {
     public JTreeExample() {
    setTitle("JTree Example");
    setSize(400, 300);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setLayout(new BorderLayout());
```

```
DefaultMutableTreeNode root = new DefaultMutableTreeNode("Root");
      DefaultMutableTreeNode folder1 = new DefaultMutableTreeNode("Folder
1");
      DefaultMutableTreeNode folder2 = new DefaultMutableTreeNode("Folder
2");
      DefaultMutableTreeNode folder3 = new DefaultMutableTreeNode("Folder
3");

      folder1.add(new DefaultMutableTreeNode("File 1-1.txt"));
      folder1.add(new DefaultMutableTreeNode("File 1-2.txt"));
      folder2.add(new DefaultMutableTreeNode("File 2-1.txt"));
      folder2.add(new DefaultMutableTreeNode("File 2-2.txt"));
      folder3.add(new DefaultMutableTreeNode("File 3-1.txt"));

      root.add(folder1);
      root.add(folder2);
      root.add(folder3);

      JTree tree = new JTree(root);
      JScrollPane treeScrollPane = new JScrollPane(tree);
      add(treeScrollPane, BorderLayout.CENTER);
   }
 public static void main(String[] args) {
      SwingUtilities.invokeLater(() -> {
         JTreeExample app = new JTreeExample();
         app.setVisible(true);
      });
   }
}
```

9. Write a Java Swing program with using Progress bar to increase the text size in the label component.

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

```
public class ProgressBarExample extends JFrame {
   private JLabel label;
   private JProgressBar progressBar;
   private JButton startButton;
   private Timer timer;
   private int fontSize = 12;
   private final int MAX_FONT_SIZE = 50;
   private final int PROGRESS_MAX = 100;

public ProgressBarExample() {
      setTitle("Progress Bar Example");
      setSize(300, 200);
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      setLayout(new FlowLayout());

      label = new JLabel("Increasing Font Size");
      label.setFont(new Font("Arial", Font.PLAIN, fontSize));
      progressBar = new JProgressBar(0, PROGRESS_MAX);
      progressBar.setStringPainted(true);

      startButton = new JButton("Start");
      startButton.addActionListener(new ActionListener() {
         @Override
         public void actionPerformed(ActionEvent e) {
            startProgress();
         }
      });
            add(label);
            add(progressBar);
            add(startButton);
   }
 private void startProgress() {
      progressBar.setValue(0);
      fontSize = 12;
      label.setFont(new Font("Arial", Font.PLAIN, fontSize));
      timer = new Timer(100, new ActionListener() {
@Override
       public void actionPerformed(ActionEvent e) {
```

```
            if (progressBar.getValue() < PROGRESS_MAX) {
            progressBar.setValue(progressBar.getValue() + 1);
            fontSize = 12 + (progressBar.getValue() * (MAX_FONT_SIZE - 12) /
PROGRESS_MAX);
            label.setFont(new Font("Arial", Font.PLAIN, fontSize));
          } else {
            timer.stop();
          }
        }
    });timer.start();
  }
public static void main(String[] args) {
    SwingUtilities.invokeLater(() -> {
      ProgressBarExample app = new ProgressBarExample();
      app.setVisible(true);
    });
  }
}
```

10. Write a Java Swing program to make a login frame using JTextField, JPasswordField and JButton class.

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

public class LoginFrame extends JFrame {
    private JTextField usernameField;
    private JPasswordField passwordField;
    private JButton loginButton;
public LoginFrame() {
     setTitle("Login Frame");
     setSize(300, 150);
     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
     setLayout(new GridLayout(3, 2));

     JLabel usernameLabel = new JLabel("Username:");
```

```
        usernameField = new JTextField();
        JLabel passwordLabel = new JLabel("Password:");
        passwordField = new JPasswordField();

        loginButton = new JButton("Login");
        loginButton.addActionListener(new ActionListener()
{
@Override
        public void actionPerformed(ActionEvent e) {
            String username = usernameField.getText();
            String password = new String(passwordField.getPassword());

           if (username.equals("admin") && password.equals("password")) {
           JOptionPane.showMessageDialog(LoginFrame.this, "Login Successful!",
"Success", JOptionPane.INFORMATION_MESSAGE);
               } else {
               JOptionPane.showMessageDialog(LoginFrame.this, "Invalid Username
or Password", "Error", JOptionPane.ERROR_MESSAGE);
               }
           }
        });
        add(usernameLabel);
        add(usernameField);
        add(passwordLabel);
        add(passwordField);
        add(new JLabel());
        add(loginButton);
    }
public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
           LoginFrame app = new LoginFrame();
           app.setVisible(true);
        });
    }
}
```

11. Write a Java Swing program to connect Student database and fetch record from the database.

Sql Program: -

```
CREATE TABLE students (
   id INT PRIMARY KEY AUTO_INCREMENT,
   name VARCHAR(100),
   age INT,
   major VARCHAR(100)
);
```

Inserting data: -

```
INSERT INTO students (name, age, major) VALUES ('Alice', 20, 'Computer
Science');
INSERT INTO students (name, age, major) VALUES ('Bob', 22, 'Mathematics');
INSERT INTO students (name, age, major) VALUES ('Charlie', 21, 'Physics');
```

Program: -

```
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.sql.*;

public class StudentDatabaseApp extends JFrame {
    private JTable table;
    private DefaultTableModel tableModel;

public StudentDatabaseApp() {
      setTitle("Student Database Records");
      setSize(400, 300);
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      setLayout(new BorderLayout());

      tableModel = new DefaultTableModel(new String[]{"ID", "Name", "Age",
"Major"}, 0);
      table = new JTable(tableModel);
      JScrollPane scrollPane = new JScrollPane(table);
      add(scrollPane, BorderLayout.CENTER);
```

```
        fetchStudentRecords();
    }
private void fetchStudentRecords() {
        String url = "jdbc:mysql://localhost:3306/your_database_name";
        String user = "your_username";
        String password = "your_password";

        try (Connection connection = DriverManager.getConnection(url, user,
password);
             Statement statement = connection.createStatement();
             ResultSet resultSet = statement.executeQuery("SELECT * FROM
students")) {

            while (resultSet.next()) {
                int id = resultSet.getInt("id");
                String name = resultSet.getString("name");
                int age = resultSet.getInt("age");
                String major = resultSet.getString("major");

                tableModel.addRow(new Object[]{id, name, age, major});
            }
        } catch (SQLException e) {
            e.printStackTrace();
            JOptionPane.showMessageDialog(this, "Error fetching data: " +
e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
        }
    }
public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            StudentDatabaseApp app = new StudentDatabaseApp();
            app.setVisible(true);
        });
    }
}
```

12. Write a Java Swing program to insert/update and delete operations from database.

SQL: -

```
CREATE TABLE students (
   id INT PRIMARY KEY AUTO_INCREMENT,
   name VARCHAR(100),
   age INT
);
```

Program: -

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

public class SimpleStudentDatabaseApp extends JFrame {
   private JTextField nameField;
   private JTextField ageField;
   private JTable table;
   private DefaultTableModel tableModel;

public SimpleStudentDatabaseApp() {
      setTitle("Simple Student Database");
      setSize(400, 300);
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      setLayout(new BorderLayout());

      JPanel inputPanel = new JPanel(new GridLayout(3, 2));
      inputPanel.add(new JLabel("Name:"));
      nameField = new JTextField();
      inputPanel.add(nameField);
      inputPanel.add(new JLabel("Age:"));
      ageField = new JTextField();
      inputPanel.add(ageField);
```

```
        JButton insertButton = new JButton("Insert");
        JButton updateButton = new JButton("Update");
        JButton deleteButton = new JButton("Delete");

        insertButton.addActionListener(e -> insertStudent());
        updateButton.addActionListener(e -> updateStudent());
        deleteButton.addActionListener(e -> deleteStudent());

        inputPanel.add(insertButton);
        inputPanel.add(updateButton);
        inputPanel.add(deleteButton);

      tableModel = new DefaultTableModel(new String[]{"ID", "Name", "Age"}, 0);
      table = new JTable(tableModel);
     JScrollPane scrollPane = new JScrollPane(table);

     add(inputPanel, BorderLayout.NORTH);
     add(scrollPane, BorderLayout.CENTER);
fetchStudentRecords();
    }
private void fetchStudentRecords() {
        String url = "jdbc:mysql://localhost:3306/your_database_name";
        String user = "your_username";
        String password = "your_password";
        try (Connection connection = DriverManager.getConnection(url, user,
password);
             Statement statement = connection.createStatement();
             ResultSet resultSet = statement.executeQuery("SELECT * FROM
students")) {
                        tableModel.setRowCount(0);
                        while (resultSet.next()) {
                        int id = resultSet.getInt("id");
                        String name = resultSet.getString("name");
                        int age = resultSet.getInt("age");
                        tableModel.addRow(new Object[]{id, name, age});
           }
        } catch (SQLException e) {
           e.printStackTrace();
```

```
            JOptionPane.showMessageDialog(this, "Error fetching data: " +
e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
        }
    }
private void insertStudent() {
        String name = nameField.getText();
        int age = Integer.parseInt(ageField.getText());
       String url = "jdbc:mysql://localhost:3306/your_database_name";
        String user = "your_username";
        String password = "your_password";
        String query = "INSERT INTO students (name, age) VALUES (?, ?)";
        try (Connection connection = DriverManager.getConnection(url, user,
password);
             PreparedStatement preparedStatement =
connection.prepareStatement(query)) {
            preparedStatement.setString(1, name);
            preparedStatement.setInt(2, age);
            preparedStatement.executeUpdate();
            fetchStudentRecords();
            clearFields();
        } catch (SQLException e) {
            e.printStackTrace();
            JOptionPane.showMessageDialog(this, "Error inserting data: " +
e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
        }
    }
private void updateStudent() {
        int selectedRow = table.getSelectedRow();
        if (selectedRow == -1) {
            JOptionPane.showMessageDialog(this, "Please select a student to
update.", "Warning ", JOptionPane.WARNING_MESSAGE);
            return;
        }
        int id = (int) tableModel.getValueAt(selectedRow, 0);
        String name = nameField.getText();
        int age = Integer.parseInt(ageField.getText());
        String url = "jdbc:mysql://localhost:3306/your_database_name";
        String user = "your_username";
```

```
        String password = "your_password";
        String query = "UPDATE students SET name = ?, age = ? WHERE id = ?";
        try (Connection connection = DriverManager.getConnection(url, user,
password);
PreparedStatement preparedStatement = connection.prepareStatement(query)) {
            preparedStatement.setString(1, name);
            preparedStatement.setInt(2, age);
            preparedStatement.setInt(3, id);
            preparedStatement.executeUpdate();
            fetchStudentRecords();
            clearFields();
        } catch (SQLException e) {
            e.printStackTrace();
            JOptionPane.showMessageDialog(this, "Error updating data: " +
e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
        }
    }
private void deleteStudent() {
        int selectedRow = table.getSelectedRow();
        if (selectedRow == -1) {
            JOptionPane.showMessageDialog(this, "Please select a student to delete.",
"Warning", JOptionPane.WARNING_MESSAGE);
            return;
        }
        int id = (int) tableModel.getValueAt(selectedRow, 0);
        String url = "jdbc:mysql://localhost:3306/your_database_name";
        String user = "your_username";
        String password = "your_password";
        String query = "DELETE FROM students WHERE id = ?";
        try (Connection connection = DriverManager.getConnection(url, user,
password);
PreparedStatement preparedStatement = connection.prepareStatement(query)) {
            preparedStatement.setInt(1, id);
            preparedStatement.executeUpdate();
            fetchStudentRecords();
            clearFields();
        } catch (SQLException e) {
            e.printStackTrace();
```

```
            JOptionPane.showMessageDialog(this, "Error deleting data: " +
e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
        }
    }
 private void clearFields() {
        nameField.setText("");
        ageField.setText("");
    }
public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            SimpleStudentDatabaseApp app = new SimpleStudentDatabaseApp();
            app.setVisible(true);
        });
    }
}
```

13. Write a java Swing program using swing to display minimum five records from the oracle table.

SQL: -

```
CREATE TABLE students (
    id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    name VARCHAR2(100),
    age NUMBER,
    major VARCHAR2(100)
);
```

Inserting data: -

```
INSERT INTO students (name, age, major) VALUES ('Alice', 20, 'Computer
Science');
INSERT INTO students (name, age, major) VALUES ('Bob', 22,
'Mathematics');
INSERT INTO students (name, age, major) VALUES ('Charlie', 21,
'Physics');
INSERT INTO students (name, age, major) VALUES ('David', 23,
'Chemistry');
INSERT INTO students (name, age, major) VALUES ('Eve', 19, 'Biology');
```

Program: -

```
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.sql.*;

public class OracleDatabaseApp extends JFrame {
   private JTable table;
   private DefaultTableModel tableModel;

public OracleDatabaseApp() {
      setTitle("Oracle Database Records");
      setSize(500, 300);
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      setLayout(new BorderLayout());

      tableModel = new DefaultTableModel(new String[]{"ID", "Name",
"Age", "Major"}, 0);
      table = new JTable(tableModel);
      JScrollPane scrollPane = new JScrollPane(table);
      add(scrollPane, BorderLayout.CENTER);

      fetchStudentRecords();
   }
private void fetchStudentRecords() {
      String url = "jdbc:oracle:thin:@localhost:1521:xe";
      String user = "your_username";
      String password = "your_password";

      try (Connection connection = DriverManager.getConnection(url, user,
password);
          Statement statement = connection.createStatement();
          ResultSet resultSet = statement.executeQuery("SELECT * FROM
students"))
{
```

```
                tableModel.setRowCount(0);
                while (resultSet.next()) {
                int id = resultSet.getInt("id");
                String name = resultSet.getString("name");
                int age = resultSet.getInt("age");
                String major = resultSet.getString("major");
                tableModel.addRow(new Object[]{id, name, age, major});
            }
        } catch (SQLException e) {
            e.printStackTrace();
            JOptionPane.showMessageDialog(this, "Error fetching data: " +
e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
        }
    }
public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            OracleDatabaseApp app = new OracleDatabaseApp();
            app.setVisible(true);
        });
    }
}
```

14. Write an XML file which will display the Book information which includes the following:

1) Title of the book 2) Author Name 3) ISBN number
4) Publisher name 5) Edition 6) Price

```
<?xml version="1.0" encoding="UTF-8"?>
<library>
  <book>
    <title>The Great Gatsby</title>
    <author>F. Scott Fitzgerald</author>
    <isbn>9780743273565</isbn>
    <publisher>Charles Scribner's Sons</publisher>
    <edition>1st</edition>
```

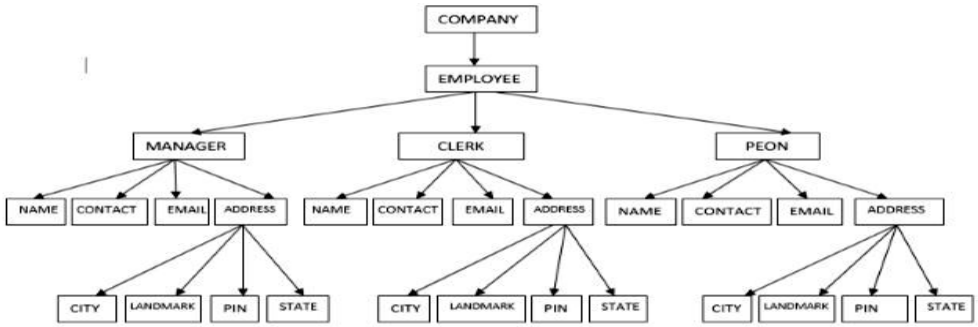
```
    <price>10.99</price>
  </book>
  <book>
    <title>To Kill a Mockingbird</title>
    <author>Harper Lee</author>
    <isbn>9780061120084</isbn>
    <publisher>J.B. Lippincott & Co.</publisher>
    <edition>1st</edition>
    <price>7.99</price>
  </book>
  <book>
    <title>1984</title>
    <author>George Orwell</author>
    <isbn>9780451524935</isbn>
    <publisher>Harcourt, Brace & Company</publisher>
    <edition>1st</edition>
    <price>9.99</price>
  </book>
  <book>
    <title>Pride and Prejudice</title>
    <author>Jane Austen</author>
    <isbn>9780141439518</isbn>
    <publisher>Penguin Classics</publisher>
    <edition>1st</edition>
    <price>8.99</price>
  </book>
  <book>
    <title>The Catcher in the Rye</title>
    <author>J.D. Salinger</author>
    <isbn>9780316769488</isbn>
    <publisher>Little, Brown and Company</publisher>
    <edition>1st</edition>
    <price>10.99</price>
  </book>
</library>
```

15. Write an XML file which will display the Student Information which includes the following:
Root element: <college>
Sub Root element: <student>
Child elements: <name>, <course>, <roll>, <session> & <address>
Sub-child elements: <city>, <state>, <pin> & <landmark>

```
<?xml version="1.0" encoding="UTF-8"?>
<college>
  <student>
    <name>John Doe</name>
    <course>Computer Science</course>
    <roll>101</roll>
    <session>2023-2024</session>
    <address>
      <city>New York</city>
      <state>NY</state>
      <pin>10001</pin>
      <landmark>Near Central Park</landmark>
    </address>
  </student>
  <student>
    <name>Jane Smith</name>
    <course>Electrical Engineering</course>
    <roll>102</roll>
    <session>2023-2024</session>
    <address>
      <city>Los Angeles</city>
      <state>CA</state>
      <pin>90001</pin>
      <landmark>Near Hollywood Sign</landmark>
    </address>
  </student>
  <student>
    <name>Emily Johnson</name>
```

```
    <course>Mechanical Engineering</course>
    <roll>103</roll>
    <session>2023-2024</session>
    <address>
      <city>Chicago</city>
      <state>IL</state>
      <pin>60601</pin>
      <landmark>Near Millennium Park</landmark>
    </address>
  </student>
  <student>
    <name>Michael Brown</name>
    <course>Civil Engineering</course>
    <roll>104</roll>
    <session>2023-2024</session>
    <address>
      <city>Houston</city>
      <state>TX</state>
      <pin>77001</pin>
      <landmark>Near Space Center</landmark>
    </address>
  </student>
  <student>
    <name>Sarah Davis</name>
    <course>Biotechnology</course>
    <roll>105</roll>
    <session>2023-2024</session>
    <address>
      <city>Miami</city>
      <state>FL</state>
      <pin>33101</pin>
      <landmark>Near South Beach</landmark>
    </address>
  </student>
</college>
```

16. Write an XML program which will display the below Employee Information which includes the following:




```
<?xml version="1.0" encoding="UTF-8"?>
<employee>
 <company>
  <employee_type>MANAGER</employee_type>
  <employee_details>
   <name></name>
   <contact></contact>
   <email></email>
   <address>
    <city></city>
    <landmark></landmark>
    <pin></pin>
    <state></state>
   </address>
  </employee_details>
 </company>
 <company>
  <employee_type>CLERK</employee_type>
  <employee_details>
   <name></name>
   <contact></contact>
   <email></email>
   <address>
    <city></city>
    <landmark></landmark>
    <pin></pin>
```

```
      <state></state>
     </address>
   </employee_details>
  </company>
  <company>
   <employee_type>PEON</employee_type>
   <employee_details>
    <name></name>
    <contact></contact>
    <email></email>
    <address>
     <city></city>
     <landmark></landmark>
     <pin></pin>
     <state></state>
    </address>
   </employee_details>
  </company>
</employee>
```

17. Write a simple JSP program to print the current date and time.

```
<%@ page import = "java.io.*,java.util.*, javax.servlet.*" %>
<html>
  <head>
    <title>Display Current Date & Time</title>
  </head>
  <body>
    <center>
      <h1>Display Current Date & Time</h1>
    </center>
    <%
      Date date = new Date();
      out.print( "<h2 align = \"center\">" +date.toString()+"</h2>");
    %>
  </body>
</html>
```

18. Write a JSP program calculates factorial values for an integer number, while the input is taken from an HTML form.

```
<html>
<body>
<form action="Factorial.jsp">
Enter a value for n: <input type="text" name="val">
<input type="submit" value="Submit">
</form>
</body>
</html>
```

Factorial.jsp

```
<html>
<body>
<%!
  long n, result;
  String str;
  long fact(long n) {
    if(n==0)
      return 1;
    else
      return n*fact(n-1);
  }
%>
<%
   str = request.getParameter("val");
   n = Long.parseLong(str);
   result = fact(n);
%>
<b>Factorial value: </b> <%= result %>
</body>
</html>
```

19. Write a JSP program calculates Powers of 2 for integers in the range 0-10. Display in tabular form (Table Row and Table Data)

```
<html>
<head>
   <title>Powers of 2</title>
</head>
<body>
<center>
<table border="2" align="center">
   <th>Exponent</th>
   <th>2^Exponent</th>
   <% for (int i=0; i<=10; i++) { //start for loop %>
        <tr>
          <td><%= i%></td>
          <td><%= Math.pow(2, i) %></td>
        </tr>
   <% } //end for loop %>
</table>
</center>
</body>
</html>
```

20. Build an application in JSP that redirects to another page.

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Redirect Page</title>
</head>
<body>
<h1>Welcome to the Redirect Page!</h1>
<p>This page will redirect to another page after 5 seconds.</p>
```

```
<script>
setTimeout(function(){
window.location.href="redirect.jsp";
}, 5000);
</script>
</body>
</html>
```

Redirect.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Redirected Page</title>
</head>
<body>
<h1>Welcome to the Redirected Page!</h1>
</body>
</html>
```