ASSIGNMENT 7

Java util Collection Frameworks and Java Swing

a) Write a program to use ArrayList to implement railway ticket booking operations.

CODE:-

```
import java.util.*;
class Ticket {
   static int counter = 1000;
   int ticketId;
   String passengerName;
   int age;
   String source;
   String destination;
   Ticket(String passengerName, int age, String source, String destination) {
      this.ticketId = counter++;
      this.passengerName = passengerName;
      this.age = age;
      this.source = source;
      this.destination = destination;
   }
   public String toString() {
      return "Ticket ID: " + ticketId + ", Passenger: " + passengerName +
            ", Age: " + age + ", From: " + source + ", To: " + destination;
   }
```

```
}
public class RailwayBooking {
    public static void main(String[] args) {
        ArrayList<Ticket> tickets = new ArrayList<>();
        Scanner sc = new Scanner(System.in);

        while (true) {
            System.out.println("\n1. Book Ticket\n2. View Tickets\n3. Cancel Ticket\n4.
Exit");
            System.out.print("Enter choice: ");
            int choice = sc.nextInt();
            sc.nextLine();

            switch (choice) {
                case 1:
                    System.out.print("Enter Name: ");
                    String name = sc.nextLine();
                    System.out.print("Enter Age: ");
                    int age = sc.nextInt();
                    sc.nextLine();
                    System.out.print("Enter Source: ");
                    String src = sc.nextLine();
                    System.out.print("Enter Destination: ");
                    String dest = sc.nextLine();

                    Ticket newTicket = new Ticket(name, age, src, dest);
                    tickets.add(newTicket);
                    System.out.println("Ticket Booked Successfully! Your Ticket ID (PNR): " +
newTicket.ticketId);
```

```
                break;

            case 2:
                if (tickets.isEmpty()) {
                    System.out.println("No tickets booked yet.");
                } else {
                    System.out.println("\nBooked Tickets:");
                    for (Ticket t : tickets)
                        System.out.println(t);
                }
                break;

            case 3:
                System.out.print("Enter Ticket ID (PNR) to cancel: ");
                int cancelId = sc.nextInt();
                boolean removed = tickets.removeIf(t -> t.ticketId == cancelId);
                if (removed)
                    System.out.println("Ticket with ID " + cancelId + " cancelled
successfully.");
                else
                    System.out.println("Ticket ID not found.");
                break;

            case 4:
                System.out.println("Exiting...");
                sc.close();
                return;
```

```
                default:
                    System.out.println("Invalid choice!");
                }
            }
        }
    }
```

---

b) Write a program to use HashMap to implment login id and password database. Use the same to verify the username and password of a given user.

CODE:-

```
import java.util.*;
public class LoginSystem {
   public static void main(String[] args) {
      HashMap<String, String> loginDB = new HashMap<>();
      Scanner sc = new Scanner(System.in);

      System.out.print("Enter number of users to add initially: ");
      int n = sc.nextInt();
      sc.nextLine();

      for (int i = 0; i < n; i++) {
         System.out.print("Enter username: ");
         String user = sc.nextLine();
         System.out.print("Enter password: ");
         String pass = sc.nextLine();
         loginDB.put(user, pass);
      }
      while (true) {
```

```
System.out.println("\n1. Add User\n2. Login\n3. View All Users\n4. Exit");
System.out.print("Enter choice: ");
int choice = sc.nextInt();
sc.nextLine();
switch (choice) {
   case 1:
      System.out.print("Enter new username: ");
      String newUser = sc.nextLine();
      if (loginDB.containsKey(newUser)) {
         System.out.println("Username already exists!");
      } else {
         System.out.print("Enter password: ");
         String newPass = sc.nextLine();
         loginDB.put(newUser, newPass);
         System.out.println("User added successfully!");
      }
      break;

   case 2:
      System.out.print("Enter username: ");
      String username = sc.nextLine();
      System.out.print("Enter password: ");
      String password = sc.nextLine();

      if (loginDB.containsKey(username)) {
         if (loginDB.get(username).equals(password)) {
            System.out.println("Login Successful! Welcome, " + username + "!");
         } else {
```

```
                System.out.println("Incorrect password!");
            }
        } else {
            System.out.println("Username not found!");
        }
        break;

    case 3:
        System.out.println("\nCurrent Users:");
        for (String user : loginDB.keySet())
            System.out.println("- " + user);
        break;

    case 4:
        System.out.println("Exiting...");
        sc.close();
        return;

    default:
        System.out.println("Invalid choice!");
    }
  }
 }
}
```

---

c) Design an application form using swing containing following fields:

i) name ,

ii) address,

iii) 10th percentage ,

iv) 12th percentage,

v) Degree percentage, and

vi) Master degree percentage.

CODE:-

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class CareerPointsForm extends JFrame {
   private JTextField nameField = new JTextField(20);
   private JTextArea addressArea = new JTextArea(3, 20);
   private JTextField t10Field = new JTextField(6);
   private JTextField t12Field = new JTextField(6);
   private JTextField degreeField = new JTextField(6);
   private JTextField masterField = new JTextField(6);
   private JLabel resultLabel = new JLabel("Total Points: 0");
   public CareerPointsForm() {
      setTitle("Application Form");
      setDefaultCloseOperation(EXIT_ON_CLOSE);
      setLayout(new BorderLayout());
      JPanel form = new JPanel(new GridBagLayout());
      GridBagConstraints c = new GridBagConstraints();
      c.insets = new Insets(6,6,6,6);
      c.anchor = GridBagConstraints.WEST;
      c.gridx = 0; c.gridy = 0; form.add(new JLabel("Name:"), c);
      c.gridx = 1; form.add(nameField, c);
      c.gridx = 0; c.gridy = 1; form.add(new JLabel("Address:"), c);
```

```
c.gridx = 1; JScrollPane sp = new JScrollPane(addressArea); form.add(sp, c);
c.gridx = 0; c.gridy = 2; form.add(new JLabel("10th Percentage:"), c);
c.gridx = 1; form.add(t10Field, c);
c.gridx = 0; c.gridy = 3; form.add(new JLabel("12th Percentage:"), c);
c.gridx = 1; form.add(t12Field, c);
c.gridx = 0; c.gridy = 4; form.add(new JLabel("Degree Percentage:"), c);
c.gridx = 1; form.add(degreeField, c);
c.gridx = 0; c.gridy = 5; form.add(new JLabel("Master Degree Percentage:"), c);
c.gridx = 1; form.add(masterField, c);

JPanel bottom = new JPanel();
JButton submit = new JButton("Submit");
bottom.add(submit);
bottom.add(resultLabel);
add(form, BorderLayout.CENTER);
add(bottom, BorderLayout.SOUTH);
submit.addActionListener(new ActionListener() {
   public void actionPerformed(ActionEvent e) {
      try {
         double p10 = parsePercent(t10Field.getText());
         double p12 = parsePercent(t12Field.getText());
         double pDeg = parsePercent(degreeField.getText());
         double pMas = parsePercent(masterField.getText());
         int total = points(p10) + points(p12) + points(pDeg) + points(pMas);
         resultLabel.setText("Total Points: " + total);
      } catch (NumberFormatException ex) {
         JOptionPane.showMessageDialog(CareerPointsForm.this, "Please enter valid
numeric percentages (0-100).", "Input Error", JOptionPane.ERROR_MESSAGE);
```

```
            }
        }
    });

    pack();
    setLocationRelativeTo(null);
    setVisible(true);
}

private double parsePercent(String s) {
    s = s.trim();
    if (s.isEmpty()) return 0.0;
    double v = Double.parseDouble(s);
    if (v < 0 || v > 100) throw new NumberFormatException();
    return v;
}
private int points(double p) {
    if (p >= 90) return 10;
    if (p >= 80) return 8;
    if (p >= 70) return 6;
    if (p >= 60) return 4;
    return 0;
}
public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            new CareerPointsForm();
        }
```

```
        });
    }
}
```

---

d) Design a small quiz form using java swing with 10 MCQs and show the points scored by the user on submit.

CODE:-

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class QuizForm extends JFrame {
    private String[] questions = {
        "1. What is encapsulation in object-oriented programming?",
        "2. Which keyword is used for class inheritance in Java?",
        "3. What does polymorphism allow in Java?",
        "4. Which statement about abstract classes in Java is true?",
        "5. What does decidability mean in theory of computation?",
        "6. What does DFA stand for?",
        "7. Which language class is accepted by a pushdown automaton?",
        "8. What is method overriding in Java?",
        "9. Which access level restricts a member to the same package (package-private)?",
        "10. The pumping lemma for regular languages is primarily used to:"
    };
    private String[][] options = {
        {"Hiding implementation details","Inheriting properties","Overloading
methods","Running code"},
        {"implement","extends","inherits","super"},
```

```
        {"Compile time binding","Different behaviors via same interface","Only single inheritance","Automatic memory management"},
        {"Can be instantiated","May contain abstract methods","Only has concrete methods","Must be final"},
        {"A problem solvable by an algorithm","Requires infinite memory","Always unsolvable","Irrelevant to computation"},
        {"Deterministic Finite Automaton","Directed Finite Algorithm","Deterministic Function Automaton","Dual Finite Automaton"},
        {"Regular languages","Context-free languages","Context-sensitive languages","Recursive languages"},
        {"Same method name, same signature in subclass","Same name but different signature in same class","Different name in subclass","Method defined only once"},
        {"private","public","protected","Default (no modifier)"},
        {"Prove a language is regular","Prove a language is not regular","Optimize finite automata","Parse context-free grammars"}
    };

    private int[] answers = {0,1,1,1,0,0,1,0,3,1};
    private JRadioButton[][] btns = new JRadioButton[10][4];
    public QuizForm() {
        setTitle("CSE Fundamentals Quiz");
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        JPanel main = new JPanel();
        main.setLayout(new BoxLayout(main, BoxLayout.Y_AXIS));
        main.setBorder(BorderFactory.createEmptyBorder(10,10,10,10));
        for (int i = 0; i < 10; i++) {
            JPanel qPanel = new JPanel();
            qPanel.setLayout(new BoxLayout(qPanel, BoxLayout.Y_AXIS));
            qPanel.setAlignmentX(Component.LEFT_ALIGNMENT);
            qPanel.add(new JLabel(questions[i]));
```

```
    ButtonGroup g = new ButtonGroup();
    for (int j = 0; j < 4; j++) {
        btns[i][j] = new JRadioButton(options[i][j]);
        g.add(btns[i][j]);
        qPanel.add(btns[i][j]);
    }
    qPanel.setBorder(BorderFactory.createEmptyBorder(8,8,8,8));
    main.add(qPanel);
}
JPanel bottom = new JPanel(new FlowLayout(FlowLayout.CENTER));
JButton submit = new JButton("Submit");
JButton reset = new JButton("Reset");
bottom.add(submit);
bottom.add(reset);
main.add(bottom);
JScrollPane sp = new JScrollPane(main);
sp.setPreferredSize(new Dimension(650, 600));
add(sp);

submit.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        int score = 0;
        for (int i = 0; i < 10; i++) {
            int selected = -1;
            for (int j = 0; j < 4; j++) {
                if (btns[i][j].isSelected()) {
                    selected = j;
                    break;
```

```
                }
            }
            if (selected == answers[i]) score++;
        }
        JOptionPane.showMessageDialog(QuizForm.this, "You scored " + score + " out of 10");
    }
  });
  reset.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
      for (int i = 0; i < 10; i++)
        for (int j = 0; j < 4; j++)
          btns[i][j].setSelected(false);
    }
  });
  pack();
  setLocationRelativeTo(null);
  setVisible(true);
 }
 public static void main(String[] args) {
  SwingUtilities.invokeLater(new Runnable() {
    public void run() {
      new QuizForm();
    }
  });
 }
}
```
