Experiment-4

Student Name: Yudhvir Singh **UID:22BCS17261**
Branch: BE-CSE **Section/Group: 22BCS_IOT-631-B**
Semester:6th **Date : 20/02/2025**
Subject Name: PBLJ **Subject Code: 22CSP-359**

Aim: Write a Java program to implement an ArrayList that stores employee details (ID, Name, and Salary). Allow users to add, update, remove, and search employees.

Code:

```
import java.util.ArrayList;
import java.util.Scanner;

class Employee {
    int id;
    String name;
    double salary;

    public Employee(int id, String name, double salary) {
        this.id = id;
        this.name = name;
        this.salary = salary;
    }
    public void displayEmployee() {
        System.out.println("ID: " + id + ", Name: " + name + ", Salary: " + salary);
    }
}
public class EmployeeManagement {
    static ArrayList<Employee> employeeList = new ArrayList<>();

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int choice;

        do {
            System.out.println("\n--- Employee Management System ---");
            System.out.println("1. Add Employee");
            System.out.println("2. Update Employee");
            System.out.println("3. Remove Employee");
            System.out.println("4. Search Employee");
            System.out.println("5. Display All Employees");
            System.out.println("6. Exit");
            System.out.print("Enter your choice: ");
            choice = scanner.nextInt();
            scanner.nextLine();

            switch (choice) {
                case 1:
                    addEmployee(scanner);
                    break;
                case 2:
                    updateEmployee(scanner);
                    break;
                case 3:
                    removeEmployee(scanner);
                    break;
                case 4:
                    searchEmployee(scanner);
                    break;
                case 5:
                    displayAllEmployees();
                    break;
                case 6:
                    System.out.println("Exiting... Thank you!");
                    break;
                default:
                    System.out.println("Invalid choice. Please try again.");
            }
        } while (choice != 6);

        scanner.close();
    }
    private static void addEmployee(Scanner scanner) {
        System.out.print("Enter Employee ID: ");
        int id = scanner.nextInt();
        scanner.nextLine();
        System.out.print("Enter Employee Name: ");
        String name = scanner.nextLine();
        System.out.print("Enter Employee Salary: ");
        double salary = scanner.nextDouble();
        Employee employee = new Employee(id, name, salary);
        employeeList.add(employee);
        System.out.println("Employee added successfully!");
    }
    private static void updateEmployee(Scanner scanner) {
        System.out.print("Enter Employee ID to update: ");
        int id = scanner.nextInt();
        scanner.nextLine();
        Employee employee = findEmployeeById(id);

        if (employee != null) {
            System.out.print("Enter new Name: ");
            String name = scanner.nextLine();
            System.out.print("Enter new Salary: ");
            double salary = scanner.nextDouble();

            employee.name = name;
            employee.salary = salary;
            System.out.println("Employee updated successfully!");
        } else {
            System.out.println("Employee not found with ID " + id);
        }
    }
    private static void removeEmployee(Scanner scanner) {
        System.out.print("Enter Employee ID to remove: ");
        int id = scanner.nextInt();

        Employee employee = findEmployeeById(id);
        if (employee != null) {
            employeeList.remove(employee);
            System.out.println("Employee removed successfully!");
        } else {
            System.out.println("Employee not found with ID " + id);
        }
    }
    private static void searchEmployee(Scanner scanner) {
        System.out.print("Enter Employee ID to search: ");
        int id = scanner.nextInt();

        Employee employee = findEmployeeById(id);
        if (employee != null) {
            employee.displayEmployee();
        } else {
            System.out.println("Employee not found with ID " + id);
        }
    }
    private static Employee findEmployeeById(int id) {
        for (Employee employee : employeeList) {
            if (employee.id == id) {
                return employee;
            }
        }
        return null;
    }
    private static void displayAllEmployees() {
        if (employeeList.isEmpty()) {
            System.out.println("No employees to display.");
        } else {
            for (Employee employee : employeeList) {
                employee.displayEmployee();
            }
        }
    }
}
```

Aim 2: Create a program to collect and store all the cards to assist the users in finding all the cards in a given symbol using Collection interface. even the head of a sorted linked list, delete all nodes that have duplicate numbers, leaving only distinct numbers from the original list. Return the linked list sorted as well.

Code:

```
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;
class Card {
    String rank;
    String suit;
    public Card(String rank, String suit) {
        this.rank = rank;
        this.suit = suit;
    }
    public String toString() {
        return rank + " of " + suit;
    }
}
public class CardDeck {
    static List<Card> deck = new ArrayList<>();
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        initializeDeck();
        int choice;
        do {
            System.out.println("\n--- Card Deck Management ---");
            System.out.println("1. Add Card");
            System.out.println("2. Display Cards by Suit");
            System.out.println("3. Display All Cards");
            System.out.println("4. Exit");
            System.out.print("Enter your choice: ");
            choice = scanner.nextInt();
            scanner.nextLine();
            switch (choice) {
                case 1:
                    addCard(scanner);
                    break;
                case 2:
                    displayCardsBySuit(scanner);
                    break;
                case 3:
                    displayAllCards();
                    break;
                case 4:
                    System.out.println("Exiting...");
                    break;
                default:
                    System.out.println("Invalid choice. Please try again.");
            }
        } while (choice != 4);
        scanner.close();
    }
private static void initializeDeck() {
        String[] suits = {"Hearts", "Diamonds", "Clubs", "Spades"};
        String[] ranks = {"2", "3", "4", "5", "6", "7", "8", "9", "10", "Jack", "Queen", "King", "Ace"};
        for (String suit : suits) {
            for (String rank : ranks) {
                deck.add(new Card(rank, suit));
            }
        }
    }
    private static void addCard(Scanner scanner) {
        System.out.print("Enter Card Rank: ");
        String rank = scanner.nextLine();
        System.out.print("Enter Card Suit: ");
        String suit = scanner.nextLine();
        deck.add(new Card(rank, suit));
        System.out.println("Card added successfully!");
    }
    private static void displayCardsBySuit(Scanner scanner) {
        System.out.print("Enter the suit to search for (Hearts, Diamonds, Clubs, Spades): ");
        String suit = scanner.nextLine();
        boolean found = false;
        for (Card card : deck) {
            if (card.suit.equalsIgnoreCase(suit)) {
                System.out.println(card);
                found = true;
            }
        }
        if (!found) {
            System.out.println("No cards found with the suit " + suit);
        }
    }
    private static void displayAllCards() {
        if (deck.isEmpty()) {
            System.out.println("No cards in the deck.");
        } else {
            for (Card card : deck) {
                System.out.println(card);
            }
        }
    }
}
```