

Name-Noorpreet Singh
Section-618-A
Subject-Java lab

4.1-

```
import java.util.*;

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

    Employee(int id, String name, double salary) {
        this.id = id;
        this.name = name;
        this.salary = salary;
    }

    @Override
    public String toString() {
        return "ID: " + id + ", Name: " + name + ", Salary: " + salary;
    }
}

public class EmployeeManagementSystem {
    private static List<Employee> employees = new ArrayList<>();

    public static void addEmployee(int id, String name, double salary) {
        Employee emp = new Employee(id, name, salary);
        employees.add(emp);
        System.out.println("Employee Added: " + emp);
    }

    public static void displayEmployees() {
        if (employees.isEmpty()) {
            System.out.println("No employees found.");
        } else {
            for (Employee emp : employees) {
                System.out.println(emp);
            }
        }
    }
}
```

```

public static void main(String[] args) {
    // Test Case 1: No Employees Initially
    System.out.println("Test Case 1: Display Employees When No
Employees Are Present");
    displayEmployees();

    // Test Case 2: Add Employees
    System.out.println("\nTest Case 2: Add Employees");
    addEmployee(101, "Anish", 50000);
    addEmployee(102, "Bobby", 60000);

    System.out.println("\nDisplaying Employees:");
    displayEmployees();
}
}

```

4.2-

```
import java.util.*;
```

```

class Card {
    String suit;
    String rank;

    Card(String suit, String rank) {
        this.suit = suit;
        this.rank = rank;
    }

    @Override
    public String toString() {
        return rank + " of " + suit;
    }
}

```

```

public class CardCollectionSystem {
    private static Map<String, List<Card>> cardMap = new HashMap<>();

    public static void addCard(String suit, String rank) {
        cardMap.putIfAbsent(suit, new ArrayList<>());
        cardMap.get(suit).add(new Card(suit, rank));
    }
}

```

```

public static void findCardsBySuit(String suit) {
    List<Card> cards = cardMap.getOrDefault(suit,
Collections.emptyList());
    if (cards.isEmpty()) {
        System.out.println("No cards found for suit: " + suit);
    } else {
        cards.forEach(System.out::println);
    }
}

public static void displayAllCards() {
    if (cardMap.isEmpty()) {
        System.out.println("No cards in the collection.");
    } else {
        cardMap.forEach((suit, cards) -> {
            System.out.println(suit + " Cards:");
            cards.forEach(System.out::println);
        });
    }
}

public static void main(String[] args) {
    // Test Case 1: No Cards Initially
    System.out.println("Test Case 1: No Cards Initially");
    System.out.println("Expected Output: No cards found.\n");
    displayAllCards();

    // Test Case 2: Adding Cards
    System.out.println("\nTest Case 2: Adding Cards");
    addCard("Spades", "Ace");
    addCard("Hearts", "King");
    addCard("Diamonds", "10");
    addCard("Clubs", "5");
    System.out.println("Expected Output:");
    System.out.println("Ace of Spades\nKing of Hearts\n10 of
Diamonds\n5 of Clubs\n");

    System.out.println("\nDisplaying All Cards:");
    displayAllCards();
}
}

```

4.3

```
import java.util.*;

class TicketBookingSystem {
    private final boolean[] seats;

    public TicketBookingSystem(int numSeats) {
        seats = new boolean[numSeats];
    }

    public synchronized boolean bookSeat(int seatNumber, String user) {
        if (seatNumber < 0 || seatNumber >= seats.length) {
            System.out.println(user + " attempted to book an invalid seat: " +
seatNumber);
            return false;
        }
        if (!seats[seatNumber]) {
            seats[seatNumber] = true;
            System.out.println(user + " successfully booked seat " +
seatNumber);
            return true;
        } else {
            System.out.println(user + " tried to book an already booked seat: "
+ seatNumber);
            return false;
        }
    }
}

class User extends Thread {
    private final TicketBookingSystem system;
    private final int seatNumber;
    private final String userType;

    public User(TicketBookingSystem system, int seatNumber, String
name, int priority) {
        super(name);
        this.system = system;
        this.seatNumber = seatNumber;
        this.userType = name;
        setPriority(priority);
    }
}
```

```

@Override
public void run() {
    system.bookSeat(seatNumber, userType);
}
}

public class TicketBookingApp {
    public static void main(String[] args) {
        int totalSeats = 5;
        TicketBookingSystem system = new
TicketBookingSystem(totalSeats);

        // Test Case 1: No Seats Available Initially
        System.out.println("Test Case 1: No Seats Available Initially");
        System.out.println("Expected Output: No bookings yet.\n");

        // Test Case 2: Successful Booking
        System.out.println("Test Case 2: Successful Booking");
        User anish = new User(system, 1, "Anish (VIP)",
Thread.MAX_PRIORITY);
        User bobby = new User(system, 2, "Bobby (Regular)",
Thread.NORM_PRIORITY);
        User charlie = new User(system, 3, "Charlie (VIP)",
Thread.MAX_PRIORITY);

        anish.start();
        bobby.start();
        charlie.start();
    }
}

```