

Rajalakshmi Engineering College

Name: Haresh R
Email: 241901031@rajalakshmi.edu.in
Roll no:
Phone: null
Branch: REC
Department: CSE (CS) - Section 1
Batch: 2028
Degree: B.E - CSE (CS)

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 5_Q4

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

You are working as a developer for CityCab, a taxi service company that wants to build a ride fare management system.

Each customer booking has:

A Booking ID (integer) A Customer Name (string) A Distance Travelled in km (double)

The fare calculation rules are:

Base Fare = 50 units (flat charge for every ride). Per km charge = 10 units/km. If the distance is greater than 20 km, a 10% discount is applied on the total fare.

You are required to implement this system using:

A class with attributes for booking details. A constructor to initialize booking details. Setter methods to update details if needed. Getter methods to retrieve details. Objects of the class to represent customer rides.

Finally, display each booking's details and final fare.

Input Format

The first line of input contains an integer N, representing the number of bookings.

For each booking:

- The next line contains the booking ID (integer).
- The following line contains the customer's name (string).
- The next line contains the distance travelled (double).

Output Format

For each booking, print the details in the following format:

1. Booking ID: <booking_id>
2. Customer Name: <customer_name>
3. Final Fare: <final_fare> (rounded to one decimal place)

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1

1234

Rahul Sharma

15

Output: Booking ID: 1234

Customer Name: Rahul Sharma

Final Fare: 200.0

Answer

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

```
class Booking {
```

```

private int bookingId;
private String customerName;
private double distance;
private double finalFare;

// Constructor
public Booking(int bookingId, String customerName, double distance) {
    this.bookingId = bookingId;
    this.customerName = customerName;
    this.distance = distance;
    this.finalFare = calculateFare();
}

// Getter methods
public int getBookingId() {
    return bookingId;
}

public String getCustomerName() {
    return customerName;
}

public double getDistance() {
    return distance;
}

public double getFinalFare() {
    return finalFare;
}

// Setter methods (if needed to update details)
public void setCustomerName(String customerName) {
    this.customerName = customerName;
}

public void setDistance(double distance) {
    this.distance = distance;
    this.finalFare = calculateFare(); // recalc if updated
}

// Fare calculation
private double calculateFare() {

```

```

        double fare = 50 + (distance * 10); // base + per km
        if (distance > 20) {
            fare = fare - (0.10 * fare); // apply 10% discount
        }
        return fare;
    }

    // Display method
    public void displayDetails() {
        System.out.printf("Booking ID: %d%n", bookingId);
        System.out.printf("Customer Name: %s%n", customerName);
        System.out.printf("Final Fare: %.1f%n", finalFare);
    }
}

class CityCabSystem {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int N = Integer.parseInt(sc.nextLine()); // number of bookings

        for (int i = 0; i < N; i++) {
            int id = Integer.parseInt(sc.nextLine());
            String name = sc.nextLine();
            double dist = Double.parseDouble(sc.nextLine());

            Booking ride = new Booking(id, name, dist);
            ride.displayDetails();
        }

        sc.close();
    }
}

```

Status : Correct

Marks : 10/10