Rajalakshmi Engineering College

Name: Sherwin G M
Email: 240701496@rajalakshmi.edu.in
Roll no: 240701496
Phone: 7708605966
Branch: REC
Department: CSE - Section 10
Batch: 2028
Degree: B.E - CSE

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 distanceTravelled;

    public Booking(int bookingId, String customerName, double distanceTravelled)
{
        this.bookingId = bookingId;
        this.customerName = customerName;
        this.distanceTravelled = distanceTravelled;
    }

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

    public void setBookingId(int bookingId) {
        this.bookingId = bookingId;
    }

    public String getCustomerName() {
        return customerName;
    }

    public void setCustomerName(String customerName) {
        this.customerName = customerName;
    }

    public double getDistanceTravelled() {
        return distanceTravelled;
    }

    public void setDistanceTravelled(double distanceTravelled) {
        this.distanceTravelled = distanceTravelled;
    }

    // Calculate fare based on the problem rules
    public double calculateFare() {
        double baseFare = 50;
        double perKmCharge = 10;
        double totalFare = baseFare + (distanceTravelled * perKmCharge);
```

```
        if (distanceTravelled > 20) {
            totalFare *= 0.90; // Apply 10% discount
        }

        return totalFare;
    }
}

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

        for (int i = 0; i < N; i++) {
            int bookingId = Integer.parseInt(sc.nextLine());
            String customerName = sc.nextLine();
            double distance = Double.parseDouble(sc.nextLine());

            Booking booking = new Booking(bookingId, customerName, distance);
            double finalFare = booking.calculateFare();

            System.out.println("Booking ID: " + booking.getBookingId());
            System.out.println("Customer Name: " + booking.getCustomerName());
            System.out.printf("Final Fare: %.1f\n", finalFare);
        }
    }
}
```

Status : Correct ***Marks : 10/10***