Rajalakshmi Engineering College

Name: rakesh kumar
Email: 241001185@rajalakshmi.edu.in
Roll no:
Phone: 6383972196
Branch: REC
Department: IT - Section 4
Batch: 2028
Degree: B.E - IT

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 5_Q5

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Ram is working as a developer for BrightEdu Coaching Center, which wants to build a student fee management system.

Each student’s enrollment has:

An Enrollment ID (integer)A Student Name (string)The Number of Subjects (integer)

The fee calculation rules are:

Registration Fee = 1000 units (flat for every student).Per Subject Fee = 800 units.If the student enrolls in more than 5 subjects, a 20% scholarship (discount) is applied on the total fee.

Ram has been asked to implement this system using:

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

Finally, display each student’s details and final fee.

Input Format

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

For each student:

- The next line contains the Enrollment ID (integer).
- The following line contains the student’s name (string).
- The next line contains the Number of subjects (integer).

Output Format

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

- Enrollment ID: <enrollment_id>
- Student Name: <student_name>
- Final Fee: <final_fee> (rounded to one decimal place)

Refer to the sample output for formatting specifications.

Sample Test Case

```
Input: 1
1234
Ravi Kumar
3
```

```
Output: Enrollment ID: 1234
Student Name: Ravi Kumar
Final Fee: 3400.0
```

Answer

```
import java.util.Scanner;

class Student {
    // Attributes
    private int enrollmentID;
    private String studentName;
    private int numberOfSubjects;

    // Constructor
    public Student(int id, String name, int subjects) {
        this.enrollmentID = id;
        this.studentName = name;
        this.numberOfSubjects = subjects;
    }

    // Setter methods
    public void setEnrollmentID(int id) {
        this.enrollmentID = id;
    }

    public void setStudentName(String name) {
        this.studentName = name;
    }

    public void setNumberOfSubjects(int subjects) {
        this.numberOfSubjects = subjects;
    }

    // Getter methods
    public int getEnrollmentID() {
        return this.enrollmentID;
    }

    public String getStudentName() {
        return this.studentName;
    }

    public int getNumberOfSubjects() {
        return this.numberOfSubjects;
    }

    // Method to calculate final fee
    public double calculateFee() {
        double fee = 1000 + (numberOfSubjects * 800); // Registration fee + per
subject fee
```

```
        // Apply 20% discount if more than 5 subjects
        if (numberOfSubjects > 5) {
            fee = fee * 0.8;
        }

        return fee;
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // Number of students
        int N = Integer.parseInt(sc.nextLine());

        // Loop through each student
        for (int i = 0; i < N; i++) {
            int id = Integer.parseInt(sc.nextLine());
            String name = sc.nextLine();
            int subjects = Integer.parseInt(sc.nextLine());

            // Create student object
            Student student = new Student(id, name, subjects);

            // Display student details
            System.out.println("Enrollment ID: " + student.getEnrollmentID());
            System.out.println("Student Name: " + student.getStudentName());
            System.out.printf("Final Fee: %.1f\n", student.calculateFee());
        }

        sc.close();
    }
}
```

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

Rajalakshmi Engineering College

Name: rakesh kumar
Email: 241001185@rajalakshmi.edu.in
Roll no:
Phone: 6383972196
Branch: REC
Department: IT - Section 4
Batch: 2028
Degree: B.E - IT

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 {
```

```
// Attributes
private int bookingID;
private String customerName;
private double distanceTravelled;

// Constructor
public Booking(int id, String name, double distance) {
    this.bookingID = id;
    this.customerName = name;
    this.distanceTravelled = distance;
}

// Setter methods
public void setBookingID(int id) {
    this.bookingID = id;
}

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

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

// Getter methods
public int getBookingID() {
    return this.bookingID;
}

public String getCustomerName() {
    return this.customerName;
}

public double getDistanceTravelled() {
    return this.distanceTravelled;
}

// Method to calculate final fare
public double calculateFare() {
    double fare = 50 + (distanceTravelled * 10); // Base fare + per km charge
```

```
        // Apply 10% discount if distance > 20 km
        if (distanceTravelled > 20) {
            fare = fare * 0.9;
        }

        return fare;
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // Number of bookings
        int N = Integer.parseInt(sc.nextLine());

        // Loop through each booking
        for (int i = 0; i < N; i++) {
            int id = Integer.parseInt(sc.nextLine());
            String name = sc.nextLine();
            double distance = Double.parseDouble(sc.nextLine());

            // Create booking object
            Booking booking = new Booking(id, name, distance);

            // Display booking details
            System.out.println("Booking ID: " + booking.getBookingID());
            System.out.println("Customer Name: " + booking.getCustomerName());
            System.out.printf("Final Fare: %.1f\n", booking.calculateFare());
        }

        sc.close();
    }
}
```

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

Rajalakshmi Engineering College

Name: rakesh kumar
Email: 241001185@rajalakshmi.edu.in
Roll no:
Phone: 6383972196
Branch: REC
Department: IT - Section 4
Batch: 2028
Degree: B.E - IT

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 5_Q3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Neha is working as a developer for CityElectricity Board, which wants to build a household electricity billing system.

Each customer’s electricity account has:

A Customer ID (integer)A Customer Name (string)Units Consumed (double)

The electricity bill is calculated based on these rules:

For the first 100 units   5 units charge per unitFor the next 100 units (101–200)   7 units charge per unitFor units above 200   10 units charge per unitIf the total bill exceeds 2000 units, a 5% discount is applied on the final bill.

Neha has been asked to implement this system using:

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

Finally, display each customer's details and final bill amount.

Input Format

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

For each customer:

- The next line contains the Customer ID (integer).
- The following line contains the Customer Name (string).
- The next line contains the Units Consumed (double).

Output Format

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

Customer ID: <customer_id>

Customer Name: <customer_name>

Final Bill: <final_bill> (rounded to one decimal place)

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1
1001
Ravi Kumar
80

Output: Customer ID: 1001
Customer Name: Ravi Kumar
Final Bill: 400.0

Answer

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

```
class Customer {
    private int customerID;
    private String customerName;
    private double unitsConsumed;
    public Customer(int id, String name, double units) {
        this.customerID = id;
        this.customerName = name;
        this.unitsConsumed = units;
    }
    public void setCustomerID(int id) {
        this.customerID = id;
    }
    public void setCustomerName(String name) {
        this.customerName = name;
    }
    public void setUnitsConsumed(double units) {
        this.unitsConsumed = units;
    }
    public int getCustomerID() {
        return this.customerID;
    }
    public String getCustomerName() {
        return this.customerName;
    }
    public double getUnitsConsumed() {
        return this.unitsConsumed;
    }
    public double calculateBill() {
        double units = this.unitsConsumed;
        double bill = 0;
        if (units <= 100) {
            bill = units * 5;
        } else if (units <= 200) {
            bill = 100 * 5 + (units - 100) * 7;
        } else {
            bill = 100 * 5 + 100 * 7 + (units - 200) * 10;
        }
        if (bill > 2000) {
            bill = bill * 0.95;
        }
        return bill;
    }
}
public class Main {
    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 id = Integer.parseInt(sc.nextLine());
            String name = sc.nextLine();
            double units = Double.parseDouble(sc.nextLine());
            Customer customer = new Customer(id, name, units);
            System.out.println("Customer ID: " + customer.getCustomerID());
            System.out.println("Customer Name: " + customer.getCustomerName());
            System.out.printf("Final Bill: %.1f\n", customer.calculateBill());
        }
        sc.close();
    }
}
```

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

Rajalakshmi Engineering College

Name: rakesh kumar
Email: 241001185@rajalakshmi.edu.in
Roll no:
Phone: 6383972196
Branch: REC
Department: IT - Section 4
Batch: 2028
Degree: B.E - IT

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 5_Q2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

You are working as a developer for CityBank, which wants to build a basic account management system.

Each customer at the bank has:

An Account Number (integer)A Customer Name (string)An Initial Balance (double)

The bank allows two types of transactions:

Deposit – increases the balance.Withdrawal – decreases the balance only if enough funds are available.

If the withdrawal amount is greater than the balance, the withdrawal should not happen, and the balance should remain the same.

You are required to implement this system using:

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

Finally, display each customer's account details after all transactions.

Input Format

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

For each customer:

- The next line contains the account number (integer).
- The following line contains the customer name (string).
- The next line contains the initial balance (double).
- The next line contains the deposit amount (double).
- The next line contains the withdrawal amount (double).

Output Format

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

1. Account Number: <account_number>
2. Customer Name: <customer_name>
3. Final Balance: <final_balance> (rounded to one decimal place)

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1
1234
Rahul Sharma
5000
2000
3000

Output: Account Number: 1234
Customer Name: Rahul Sharma

```
Final Balance: 4000.0
```

Answer

```
import java.util.Scanner;

class Account {
    private int accountNumber;
    private String customerName;
    private double balance;

    public Account(int accountNumber, String customerName, double balance) {
        this.accountNumber = accountNumber;
        this.customerName = customerName;
        this.balance = balance;
    }

    public void deposit(double amount) {
        if (amount >= 0) balance += amount;
    }

    public void withdraw(double amount) {
        if (amount >= 0 && amount <= balance) balance -= amount;
    }

    public int getAccountNumber() { return accountNumber; }
    public String getCustomerName() { return customerName; }
    public double getBalance() { return balance; }
}

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

        for (int i = 0; i < N; i++) {
            int accNo = Integer.parseInt(sc.nextLine().trim());
            String name = sc.nextLine();
            double initialBalance = Double.parseDouble(sc.nextLine().trim());
            double depositAmount = Double.parseDouble(sc.nextLine().trim());
            double withdrawAmount = Double.parseDouble(sc.nextLine().trim());

            Account customer = new Account(accNo, name, initialBalance);
            customer.deposit(depositAmount);
            customer.withdraw(withdrawAmount);

            System.out.println("Account Number: " + customer.getAccountNumber());
            System.out.println("Customer Name: " + customer.getCustomerName());
            System.out.printf("Final Balance: %.1f%n", customer.getBalance());
        }
        sc.close();
    }
}
```

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