

Rajalakshmi Engineering College

Name: Devyesh C
Email: 241801049@rajalakshmi.edu.in
Roll no: 241801049
Phone: 7708811709
Branch: REC
Department: AI & DS - Section 5
Batch: 2028
Degree: B.E - AI & DS

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 Citybank{
    int acc_no;
    String cus_name;
    double init_bal;
    Citybank(int acno, String cn, double bal){
        this.acc_no = acno;
        this.cus_name = cn;
        this.init_bal = bal;
    }
    void deposit(double amt){
        init_bal += amt;
    }
    void withdraw(double amt){
        if(amt<=init_bal){
            init_bal -= amt;
        }
    }
    void display_details(){
        System.out.println("Account Number: "+ acc_no);
        System.out.println("Customer Name: "+ cus_name);
        System.out.printf("Final Balance: %.1f\n", init_bal);
    }
}

class Main{
    public static void main(String[] args){
        int N;
        Scanner sc = new Scanner(System.in);
        N = Integer.parseInt(sc.nextLine());
        for(int i = 0; i<N; i++){
            int acc_no = Integer.parseInt(sc.nextLine());
            String cus_name = sc.nextLine();
            double init_bal = Double.parseDouble(sc.nextLine());
```

```
        Citybank ob = new Citybank(acc_no, cus_name, init_bal);
        double dp_amt = Double.parseDouble(sc.nextLine());
        ob.deposit(dp_amt);
        double wd_amt = Double.parseDouble(sc.nextLine());
        ob.withdraw(wd_amt);
        ob.display_details();
    }
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: Devyesh C
Email: 241801049@rajalakshmi.edu.in
Roll no: 241801049
Phone: 7708811709
Branch: REC
Department: AI & DS - Section 5
Batch: 2028
Degree: B.E - AI & DS

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 unit
For the next 100 units (101–200) 7 units charge per unit
For units above 200 10 units charge per unit
If 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 Electricity_account{  
    int cus_id;  
    String cus_name;  
    double units_cons;  
    double final_bill;  
    Electricity_account(int cus_id, String cus_name, double units_cons){  
        this.cus_id = cus_id;  
        this.cus_name = cus_name;  
        this.units_cons = units_cons;  
    }  
  
    void calculate_bill(){  
        if(units_cons<=100){  
            final_bill = units_cons*5;  
        }  
        else if(units_cons>100 && units_cons<=200){  
            final_bill = (100*5)+(units_cons-100)*7;  
        }  
        else if(units_cons>200){  
            final_bill = (100*5)+(100*7)+(units_cons-200)*10;  
        }  
  
        if(final_bill>2000){  
            final_bill *= 0.95;  
        }  
    }  
  
    void display(){  
        System.out.println("Customer ID: "+cus_id);  
        System.out.println("Customer Name: "+ cus_name);  
        System.out.printf("Final Bill: %.1f\n", final_bill);  
    }  
}  
  
class Main{  
    public static void start(){  
        Scanner sc = new Scanner(System.in);  
        int T = Integer.parseInt(sc.nextLine());  
        for(int i = 0; i<T; i++){  
            int id = Integer.parseInt(sc.nextLine());  
            String name = sc.nextLine();  
        }  
    }  
}
```

```
        double units = Double.parseDouble(sc.nextLine());
        Electricity_account obj = new Electricity_account(id, name, units);
        obj.calculate_bill();
        obj.display();
    }

public static void main(String[] args){
    start();
}
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: Devyesh C
Email: 241801049@rajalakshmi.edu.in
Roll no: 241801049
Phone: 7708811709
Branch: REC
Department: AI & DS - Section 5
Batch: 2028
Degree: B.E - AI & DS

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

```
// You are using Java  
import java.util.Scanner;
```

```
class Booking {  
    private int bookingId;  
    private String customerName;  
    private double distance;  
    private double fare;  
  
    public Booking(int bookingId, String customerName, double distance) {  
        this.bookingId = bookingId;  
        this.customerName = customerName;  
        this.distance = distance;  
        calculateFare();  
    }  
  
    public void setBookingId(int bookingId) {  
        this.bookingId = bookingId;  
    }  
  
    public void setCustomerName(String customerName) {  
        this.customerName = customerName;  
    }  
  
    public void setDistance(double distance) {  
        this.distance = distance;  
        calculateFare();  
    }  
  
    public int getBookingId() {  
        return bookingId;  
    }  
  
    public String getCustomerName() {  
        return customerName;  
    }  
  
    public double getDistance() {  
        return distance;  
    }  
  
    public double getFare() {  
        return fare;  
    }  
}
```

```
private void calculateFare() {  
    fare = 50 + distance * 10;  
    if (distance > 20) {  
        fare = fare * 0.90;  
    }  
}  
  
class CityCabApp {  
    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 distance = Double.parseDouble(sc.nextLine());  
            Booking booking = new Booking(id, name, distance);  
            System.out.println("Booking ID: " + booking.getBookingId());  
            System.out.println("Customer Name: " + booking.getCustomerName());  
            System.out.printf("Final Fare: %.1f\n", booking.getFare());  
        }  
        sc.close();  
    }  
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: Devyesh C
Email: 241801049@rajalakshmi.edu.in
Roll no: 241801049
Phone: 7708811709
Branch: REC
Department: AI & DS - Section 5
Batch: 2028
Degree: B.E - AI & DS

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_enrollment{  
    int enrol_id;
```

```
String stud_name;
int no_of_sub;
double fee;
Student_enrollment(int enrol_id, String stud_name, int no_of_sub){
    this.enrol_id = enrol_id;
    this.stud_name = stud_name;
    this.no_of_sub = no_of_sub;
    fee_calculation();
}

void fee_calculation(){
    fee = 1000+(800*no_of_sub);
    if(no_of_sub>5){
        fee = fee*0.80;
    }
}

int get_id(){
    return enrol_id;
}

String get_name(){
    return stud_name;
}

double get_fee(){
    return fee;
}

class Main{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        int T;
        T = Integer.parseInt(sc.nextLine());
        for(int i = 0; i<T; i++){
            int id = Integer.parseInt(sc.nextLine());
            String name = sc.nextLine();
            int no_sub = Integer.parseInt(sc.nextLine());
            Student_enrollment obj = new Student_enrollment(id, name, no_sub);
            System.out.println("Enrollment ID: "+ obj.get_id());
            System.out.println("Student Name: "+ obj.get_name());
        }
    }
}
```

```
        } } System.out.printf("Final Fee: %.1f", obj.get_fee());  
    } }
```

Status : Correct

Marks : 10/10