

# Rajalakshmi Engineering College

Name: Jathin Chowdary CH  
Email: 240701209@rajalakshmi.edu.in  
Roll no: 240701209  
Phone: 8939767767  
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\_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 Customer {  
    private int customerId;  
    private String customerName;  
    private double units;  
  
    Customer(int customerId, String customerName, double units) {  
        this.customerId = customerId;  
        this.customerName = customerName;  
        this.units = units;  
    }  
  
    public int getCustomerId() {  
        return customerId;  
    }  
  
    public String getCustomerName() {  
        return customerName;  
    }  
  
    public double getUnits() {  
        return units;  
    }  
  
    public void setCustomerId(int customerId) {  
        this.customerId = customerId;  
    }  
  
    public void setCustomerName(String customerName) {  
        this.customerName = customerName;  
    }  
  
    public void setUnits(double units) {  
        this.units = units;  
    }  
  
    public double calculateBill() {  
        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) * 9;  
        }  
        return bill;  
    }  
}
```

```
        } else {
            bill = 100 * 5 + 100 * 7 + (units - 200) * 10;
        }
        if (bill > 2000) {
            bill = bill - (bill * 0.05);
        }
        return bill;
    }
}

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 c = new Customer(id, name, units);
            System.out.println("Customer ID: " + c.getCustomerId());
            System.out.println("Customer Name: " + c.getCustomerName());
            System.out.println("Final Bill: " + String.format("%.1f", c.calculateBill()));
        }
    }
}
```

**Status : Correct**

**Marks : 10/10**