Rajalakshmi Engineering College

Name: varsha s
Email: 241501237@rajalakshmi.edu.in
Roll no:
Phone: 9342191041
Branch: REC
Department: AI & ML - Section 1
Batch: 2028
Degree: B.E - AI & ML

Scan to verify results

2024_28_III_OOPS Using Java Lab

REC_2028_OOPS using Java_Week 5_CY

Attempt : 1
Total Mark : 40
Marks Obtained : 40

Section 1 : Coding

1. Problem Statement

Arjun is working as a developer for CityWater Supply Board, which wants to build a household water billing system.

Each household's water account has:

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

The water bill is calculated based on these rules:

For the first 500 liters 2 per literFor the next 500 liters (501–1000) 3 per literFor liters above 1000 5 per literIf the total bill exceeds 3000, a 10% discount is applied on the final bill.

Arjun 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 Liters 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
300
Output: Customer ID: 1001
Customer Name: Ravi Kumar
Final Bill: 600.0

Answer

// You are using Java

```
import java.util.Scanner;

class Customer {
    private int customerId;
    private String customerName;
    private double litersConsumed;

    public Customer(int customerId, String customerName, double
litersConsumed) {
        this.customerId = customerId;
        this.customerName = customerName;
        this.litersConsumed = litersConsumed;
    }

    // Getters
    public int getCustomerId() {
        return customerId;
    }

    public String getCustomerName() {
        return customerName;
    }

    public double getLitersConsumed() {
        return litersConsumed;
    }

    // Setters (if needed)
    public void setCustomerId(int customerId) {
        this.customerId = customerId;
    }

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

    public void setLitersConsumed(double litersConsumed) {
        this.litersConsumed = litersConsumed;
    }

    public double calculateBill() {
        double liters = litersConsumed;
```

```
        double bill = 0.0;

        // First 500 liters at   2/liter
        double tier1 = Math.min(liters, 500);
        bill += tier1 * 2;
        liters -= tier1;

        if (liters > 0) {
            // Next 500 liters (501–1000) at   3/liter
            double tier2 = Math.min(liters, 500);
            bill += tier2 * 3;
            liters -= tier2;
        }

        if (liters > 0) {
            // Above 1000 liters at   5/liter
            bill += liters * 5;
        }

        // If total bill exceeds   3000, apply 10% discount
        if (bill > 3000) {
            bill = bill * 0.90;  // 10% discount
        }

        return bill;
    }
}

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

        int N = Integer.parseInt(sc.nextLine().trim());
        Customer[] customers = new Customer[N];

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

            customers[i] = new Customer(id, name, liters);
        }
```

```
        for (Customer c : customers) {
            double finalBill = c.calculateBill();
            System.out.printf("Customer ID: %d%n", c.getCustomerId());
            System.out.printf("Customer Name: %s%n", c.getCustomerName());
            System.out.printf("Final Bill: %.1f%n", finalBill);
        }

        sc.close();
    }
}
```

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

2. Problem Statement

You are working as a developer for CityMobile, which wants to build a basic mobile data usage management system.

Each customer has:

A Customer ID (integer)A Customer Name (string)An Initial Data Balance (in GB, double)

The company allows two types of operations:

Recharge – increases the data balance.Usage – decreases the data balance only if enough data is available.

If the usage amount is greater than the available data balance, the usage should not happen, and the balance should remain the same.

You are required 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 after all operations.

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 Initial Data Balance (double).
- The next line contains the Recharge Amount in GB (double).
- The next line contains the Usage Amount in GB (double).

Output Format

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

Customer ID: <customer_id>

Customer Name: <customer_name>

Final Data Balance: <final_data_balance> GB (The final balance must be rounded to one decimal place.)

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1
1234
Ravi Kumar
5.0
2.0
3.0
Output: Customer ID: 1234
Customer Name: Ravi Kumar
Final Data Balance: 4.0 GB

Answer

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

class Customer {
```

```
    private int customerId;
    private String customerName;
    private double dataBalance;  // in GB

    public Customer(int customerId, String customerName, double initialBalance)
{
        this.customerId = customerId;
        this.customerName = customerName;
        this.dataBalance = initialBalance;
    }

    // Getters
    public int getCustomerId() {
        return customerId;
    }

    public String getCustomerName() {
        return customerName;
    }

    public double getDataBalance() {
        return dataBalance;
    }

    // Setters (if needed)
    public void setCustomerId(int customerId) {
        this.customerId = customerId;
    }

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

    public void setDataBalance(double dataBalance) {
        this.dataBalance = dataBalance;
    }

    // Recharge method: increases balance
    public void recharge(double amount) {
        if (amount < 0) {
            // ignore invalid negative recharge
            return;
```

```
        }
        this.dataBalance += amount;
    }

    // Usage method: decreases balance if enough data is available
    public void useData(double amount) {
        if (amount < 0) {
            // ignore invalid negative usage
            return;
        }
        if (amount <= this.dataBalance) {
            this.dataBalance -= amount;
        } else {
            // If usage is more than balance, usage is ignored
        }
    }
}

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

        int N = Integer.parseInt(sc.nextLine().trim());
        Customer[] customers = new Customer[N];

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

            Customer c = new Customer(id, name, initialBal);
            c.recharge(rechargeAmt);
            c.useData(usageAmt);

            customers[i] = c;
        }

        for (Customer c : customers) {
            System.out.printf("Customer ID: %d%n", c.getCustomerId());
            System.out.printf("Customer Name: %s%n", c.getCustomerName());
```

```
            System.out.printf("Final Data Balance: %.1f GB%n", c.getDataBalance());
        }

        sc.close();
    }
}
```

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

3. Problem Statement

Meera is working as a developer for CityGas Supply Board, which wants to build a household gas billing system.

Each household’s gas account has:

A Customer ID (integer)A Customer Name (string)Units Consumed in cubic meters (double)

The gas bill is calculated based on these rules:

For the first 50 units 4 per unitFor the next 100 units (51–150) 6 per unitFor units above 150 8 per unitIf the total bill exceeds 2000, a 15% discount is applied on the final bill.

Meera 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> (The final bill must be rounded to one decimal place.)

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1
1001
Ravi Kumar
30
Output: Customer ID: 1001
Customer Name: Ravi Kumar
Final Bill: 120.0

Answer

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

class Customer {
   private int customerId;
   private String customerName;
   private double unitsConsumed;

   public Customer(int customerId, String customerName, double
unitsConsumed) {
       this.customerId = customerId;
       this.customerName = customerName;
       this.unitsConsumed = unitsConsumed;
   }
```

```
// Getters
public int getCustomerId() {
    return customerId;
}

public String getCustomerName() {
    return customerName;
}

public double getUnitsConsumed() {
    return unitsConsumed;
}

// Setters (if needed)
public void setCustomerId(int customerId) {
    this.customerId = customerId;
}

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

public void setUnitsConsumed(double unitsConsumed) {
    this.unitsConsumed = unitsConsumed;
}

public double calculateBill() {
    double units = unitsConsumed;
    double bill = 0.0;

    // First 50 units at   4 per unit
    double tier1 = Math.min(units, 50);
    bill += tier1 * 4;
    units -= tier1;

    if (units > 0) {
        // Next 100 units (51–150) at   6 per unit
        double tier2 = Math.min(units, 100);
        bill += tier2 * 6;
        units -= tier2;
    }
```

```
        if (units > 0) {
            // Units above 150 at   8 per unit
            bill += units * 8;
        }

        // If the total bill exceeds   2000, apply 15% discount
        if (bill > 2000) {
            bill = bill * 0.85;  // 15% off
        }

        return bill;
    }
}


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

        int N = Integer.parseInt(sc.nextLine().trim());
        Customer[] customers = new Customer[N];

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

            customers[i] = new Customer(id, name, units);
        }

        for (Customer c : customers) {
            double finalBill = c.calculateBill();
            System.out.printf("Customer ID: %d%n", c.getCustomerId());
            System.out.printf("Customer Name: %s%n", c.getCustomerName());
            System.out.printf("Final Bill: %.1f%n", finalBill);
        }

        sc.close();
    }
}
```

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

4. Problem Statement

Anjali is now working as a developer for the City Marathon Association, which wants to build a system to track and find the fastest runner among marathon participants.

Each runner’s record has:

Runner ID (integer)Runner Name (string)An array of times (in minutes) taken in 5 marathon events (integers)

The system must calculate:

The average time of each runner (sum of all times / 5).Identify the fastest runner (the one with the lowest average time).If two or more runners have the same average time, the one with the lower Runner ID is considered the fastest runner.

Anjali has been asked to implement this system using:

A class with attributes for runner details.A constructor to initialize runner details.Getter and Setter methods to retrieve and update runner details if required.A method to calculate the average time.Objects of the class to represent runners.

Finally, display each runner’s details and announce the Fastest Runner.

Input Format

The first line of input contains an integer N (number of runners).

For each runner:

- The next line contains the Runner ID (integer).
- The following line contains the Runner Name (string).
- The next line contains 5 integers separated by spaces (times in minutes for 5 marathon events).

Output Format

For each runner the output prints the following details:

- Runner ID: <runner_id>

- Runner Name: <runner_name>
- Average Time: <average_time>

Finally, print "Fastest Runner: <runner_name> with <average_time> minutes"

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1
1001
Ravi Kumar
240 250 245 255 260

Output: Runner ID: 1001
Runner Name: Ravi Kumar
Average Time: 250
Fastest Runner: Ravi Kumar with 250 minutes

Answer

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

class Runner {
    private int runnerId;
    private String runnerName;
    private int[] times;  // times for 5 events

    public Runner(int runnerId, String runnerName, int[] times) {
        this.runnerId = runnerId;
        this.runnerName = runnerName;
        this.times = times;
    }

    // Getters and setters
    public int getRunnerId() {
        return runnerId;
    }
```

```
    public String getRunnerName() {
        return runnerName;
    }

    public int[] getTimes() {
        return times;
    }

    public void setRunnerId(int runnerId) {
        this.runnerId = runnerId;
    }

    public void setRunnerName(String runnerName) {
        this.runnerName = runnerName;
    }

    public void setTimes(int[] times) {
        this.times = times;
    }

    public int averageTime() {
        int sum = 0;
        for (int t : times) {
            sum += t;
        }
        return sum / times.length;  // integer division; times.length is 5
    }
}

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

        int N = Integer.parseInt(sc.nextLine().trim());
        Runner[] runners = new Runner[N];

        for (int i = 0; i < N; i++) {
            int id = Integer.parseInt(sc.nextLine().trim());
            String name = sc.nextLine().trim();
            String[] parts = sc.nextLine().trim().split("\\s+");
            if (parts.length != 5) {
```

```
            // input error, but assume valid inputs as per specification
        }
        int[] times = new int[5];
        for (int j = 0; j < 5; j++) {
            times[j] = Integer.parseInt(parts[j]);
        }
        runners[i] = new Runner(id, name, times);
    }

    // Print each runner and track fastest
    int bestAvg = Integer.MAX_VALUE;
    int bestRunnerIndex = -1;

    for (int i = 0; i < N; i++) {
        Runner r = runners[i];
        int avg = r.averageTime();
        System.out.printf("Runner ID: %d%n", r.getRunnerId());
        System.out.printf("Runner Name: %s%n", r.getRunnerName());
        System.out.printf("Average Time: %d%n", avg);

        if (avg < bestAvg) {
            bestAvg = avg;
            bestRunnerIndex = i;
        } else if (avg == bestAvg) {
            // tie    lower runner ID wins
            if (r.getRunnerId() < runners[bestRunnerIndex].getRunnerId()) {
                bestRunnerIndex = i;
            }
        }
    }

    if (bestRunnerIndex >= 0) {
        Runner best = runners[bestRunnerIndex];
        System.out.printf("Fastest Runner: %s with %d minutes%n",
                          best.getRunnerName(), best.averageTime());
    }

    sc.close();
  }
}
```

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