

Rajalakshmi Engineering College

Name: Gayathri Boopathy
Email: 240701141@rajalakshmi.edu.in
Roll no: 240701141
Phone: 9363837860
Branch: REC
Department: CSE - Section 10
Batch: 2028
Degree: B.E - CSE

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

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

        for (int i = 0; i < N; i++) {
            int id = Integer.parseInt(scanner.nextLine());
            String name = scanner.nextLine();
            double balance = Double.parseDouble(scanner.nextLine());
            double recharge = Double.parseDouble(scanner.nextLine());
            double usage = Double.parseDouble(scanner.nextLine());

            Customer customer = new Customer(id, name, balance);
            customer.rechargeData(recharge);
            customer.useData(usage);

            System.out.println("Customer ID: " + customer.getId());
            System.out.println("Customer Name: " + customer.getName());
            System.out.printf("Final Data Balance: %.1f GB%n",
                customer.getBalance());
        }
        scanner.close();
    }
}

class Customer {
    private int id;
    private String name;
    private double dataBalance;

    public Customer(int id, String name, double balance) {
        this.id = id;
        this.name = name;
    }
}
```

```
        this.dataBalance = balance;
    }

    public void rechargeData(double amount) {
        if (amount > 0) {
            dataBalance += amount;
        }
    }

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

    public int getId() { return id; }
    public String getName() { return name; }
    public double getBalance() {
        return Math.round(dataBalance * 10) / 10.0;
    }

    public void setId(int id) { this.id = id; }
    public void setName(String name) { this.name = name; }
    public void setBalance(double balance) { this.dataBalance = balance; }
}
```

Status : Correct

Marks : 10/10

2. 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 liter
For 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 WaterBillingSystem {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int N = Integer.parseInt(scanner.nextLine());

        for (int i = 0; i < N; i++) {
            int customerId = Integer.parseInt(scanner.nextLine());
            String customerName = scanner.nextLine();
            double liters = Double.parseDouble(scanner.nextLine());

            Customer customer = new Customer(customerId, customerName, liters);
            double finalBill = customer.calculateBill();

            System.out.println("Customer ID: " + customer.getId());
            System.out.println("Customer Name: " + customer.getName());
            System.out.printf("Final Bill: %.1f%n", finalBill);
        }
        scanner.close();
    }
}

class Customer {
    private int id;
    private String name;
    private double litersConsumed;

    public Customer(int id, String name, double liters) {
        this.id = id;
        this.name = name;
        this.litersConsumed = liters;
    }

    public double calculateBill() {
        double remaining = litersConsumed;
        double total = 0.0;
```

```

        if (remaining > 1000) {
            total += (remaining - 1000) * 5;
            remaining = 1000;
        }
        if (remaining > 500) {
            total += (remaining - 500) * 3;
            remaining = 500;
        }
        total += remaining * 2;

        if (total > 3000) {
            total *= 0.9;
        }
    return Math.round(total * 10) / 10.0;
}

// Getters and setters
public int getId() { return id; }
public String getName() { return name; }
public double getLiters() { return litersConsumed; }

public void setId(int id) { this.id = id; }
public void setName(String name) { this.name = name; }
public void setLiters(double liters) { this.litersConsumed = liters; }
}

```

Status : Correct

Marks : 10/10

3. Problem Statement

Anjali is working as a developer for the City Basketball Association, which wants to build a system to track and find the top scorer among basketball players.

Each player's record has:

Player ID (integer) Player Name (string) An array of points scored in 5 matches (integers)

The system must calculate:

The total score of each player (sum of all match points). Identify the highest scorer among all players. If two or more players have the same total score, the one with the lower Player ID is considered the top scorer.

Anjali has been asked to implement this system using:

A class with attributes for player details. A constructor to initialize player details. Getter and Setter methods to retrieve and update player details if required. A method to calculate the total score. Objects of the class to represent players.

Finally, display each player's details and announce the Top Scorer.

Input Format

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

For each player:

- The next line contains the Player ID (integer).
- The following line contains the Player Name (string).
- The next line contains 5 integers separated by spaces (points scored in 5 matches).

Output Format

For each player the output prints the following details:

- Player ID: <player_id>
- Player Name: <player_name>
- Total Score: <total_score>

Finally, print "Top Scorer: <player_name> with <total_score> points"

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1
1001
Ravi Kumar
10 20 30 40 50

Output: Player ID: 1001
Player Name: Ravi Kumar
Total Score: 150
Top Scorer: Ravi Kumar with 150 points

Answer

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

class BasketballScorerSystem {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        ArrayList<Player> players = new ArrayList<>();
        int N = Integer.parseInt(scanner.nextLine());

        for (int i = 0; i < N; i++) {
            int id = Integer.parseInt(scanner.nextLine());
            String name = scanner.nextLine();
            String[] pointsStr = scanner.nextLine().split(" ");
            int[] points = new int[5];
            for (int j = 0; j < 5; j++) {
                points[j] = Integer.parseInt(pointsStr[j]);
            }
            players.add(new Player(id, name, points));
        }

        Player topScorer = players.get(0);
        for (Player p : players) {
            System.out.println("Player ID: " + p.getId());
            System.out.println("Player Name: " + p.getName());
            System.out.println("Total Score: " + p.getTotalScore());

            if (p.getTotalScore() > topScorer.getTotalScore() ||
                (p.getTotalScore() == topScorer.getTotalScore() && p.getId() <
topScorer.getId())) {
```

```
        topScorer = p;
    }
}

System.out.println("Top Scorer: " + topScorer.getName() + " with " +
topScorer.getTotalScore() + " points");
scanner.close();
}
}
```

```
class Player {
    private int id;
    private String name;
    private int[] points;
    private int totalScore;

    public Player(int id, String name, int[] points) {
        this.id = id;
        this.name = name;
        this.points = points;
        calculateTotalScore();
    }

    private void calculateTotalScore() {
        totalScore = 0;
        for (int score : points) {
            totalScore += score;
        }
    }

    public int getId() { return id; }
    public String getName() { return name; }
    public int getTotalScore() { return totalScore; }
```

```
    public void setId(int id) { this.id = id; }
    public void setName(String name) { this.name = name; }
    public void setPoints(int[] points) {
        this.points = points;
        calculateTotalScore();
    }
}
```

4. 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 GasBillingSystem {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int N = Integer.parseInt(scanner.nextLine());

        for (int i = 0; i < N; i++) {
            int customerId = Integer.parseInt(scanner.nextLine());
            String customerName = scanner.nextLine();
            double units = Double.parseDouble(scanner.nextLine());

            Customer customer = new Customer(customerId, customerName, units);
            double finalBill = customer.calculateBill();

            System.out.println("Customer ID: " + customer.getId());
            System.out.println("Customer Name: " + customer.getName());
            System.out.printf("Final Bill: %.1f%n", finalBill);
        }
        scanner.close();
    }
}
```

```
}

class Customer {
    private int id;
    private String name;
    private double unitsConsumed;

    public Customer(int id, String name, double units) {
        this.id = id;
        this.name = name;
        this.unitsConsumed = units;
    }

    public double calculateBill() {
        double remaining = unitsConsumed;
        double total = 0.0;

        if (remaining > 150) {
            total += (remaining - 150) * 8;
            remaining = 150;
        }
        if (remaining > 50) {
            total += (remaining - 50) * 6;
            remaining = 50;
        }
        total += remaining * 4;

        if (total > 2000) {
            total *= 0.85;
        }

        return Math.round(total * 10) / 10.0;
    }

    // Getters and setters
    public int getId() { return id; }
    public String getName() { return name; }
    public double getUnits() { return unitsConsumed; }

    public void setId(int id) { this.id = id; }
    public void setName(String name) { this.name = name; }
    public void setUnits(double units) { this.unitsConsumed = units; }
}
```

}

Status : Correct

Marks : 10/10