Rajalakshmi Engineering College

Name: KRITHESHWARAN R
Email: 241901049@rajalakshmi.edu.in
Roll no: 241901049
Phone: 9843565002
Branch: REC
Department: CSE (CS) - Section 2
Batch: 2028
Degree: B.E - CSE (CS)

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

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

```java
import java.util.Scanner;

class Player {
    private int id;
    private String name;
    private int[] scores = new int[5];
    private int total;

    public Player(int id, String name, int[] scores) {
        this.id = id;
        this.name = name;
        this.scores = scores;
        this.total = 0;
        for (int i = 0; i < 5; i++) {
            this.total += scores[i];
        }
    }

    public int getId() {
        return id;
    }

    public String getName() {
        return name;
    }

    public int getTotal() {
        return total;
    }
}

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

```java
    Player[] players = new Player[n];

    int topScore = -1;
    int topId = 9999;
    String topName = "";

    for (int i = 0; i < n; i++) {
        int id = sc.nextInt();
        sc.nextLine();
        String name = sc.nextLine();
        int[] scores = new int[5];
        int total = 0;
        for (int j = 0; j < 5; j++) {
            scores[j] = sc.nextInt();
            total += scores[j];
        }
        //sc.nextLine();
        players[i] = new Player(id, name, scores);

        System.out.println("Player ID: " + players[i].getId());
        System.out.println("Player Name: " + players[i].getName());
        System.out.println("Total Score: " + players[i].getTotal());

        if (players[i].getTotal() > topScore) {
            topScore = players[i].getTotal();
            topId = players[i].getId();
            topName = players[i].getName();
        } else if (players[i].getTotal() == topScore && players[i].getId() < topId) {
            topId = players[i].getId();
            topName = players[i].getName();
        }
    }

    System.out.println("Top Scorer: " + topName + " with " + topScore + "
points");
    }
}
```

Status : Correct   *Marks : 10/10*

2. 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

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

class Customer {
    private int id;
    private String name;
    private double units;
    private double bill;

    public Customer(int id, String name, double units) {
        this.id = id;
        this.name = name;
        this.units = units;
        this.bill = 0.0;
    }

    public int getId() {
        return id;
    }

    public String getName() {
        return name;
    }

    public double getUnits() {
        return units;
    }
}
```

```java
    public double getBill() {
        return bill;
    }

    public void setBill(double bill) {
        this.bill = bill;
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        sc.nextLine();
        Customer[] customers = new Customer[n];

        for (int i = 0; i < n; i++) {
            int id = sc.nextInt();
            sc.nextLine();
            String name = sc.nextLine();
            double units = sc.nextDouble();

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

            double bill = 0.0;

            if (units <= 50) {
                bill = units * 4;
            } else if (units <= 150) {
                bill = (50 * 4) + ((units - 50) * 6);
            } else {
                bill = (50 * 4) + (100 * 6) + ((units - 150) * 8);
            }

            if (bill > 2000) {
                bill = bill - (bill * 0.15);
            }
            customers[i].setBill(bill);

            System.out.println("Customer ID: " + customers[i].getId());
            System.out.println("Customer Name: " + customers[i].getName());
```

```
        System.out.println("Final Bill: " + String.format("%.1f",
customers[i].getBill()));
        }
    }
}
```

Status : Correct   *Marks : 10/10*

3. 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 id;
    private String name;
    private int[] times = new int[5];
    private int average;
```

```java
    public Runner(int id, String name, int[] times) {
        this.id = id;
        this.name = name;
        this.times = times;
        calculateAverage();
    }

    public void calculateAverage() {
        int sum = 0;
        for (int i = 0; i < 5; i++) {
            sum += times[i];
        }
        this.average = sum / 5;
    }

    public int getId() {
        return id;
    }

    public String getName() {
        return name;
    }

    public int getAverage() {
        return average;
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        sc.nextLine();
        Runner[] runners = new Runner[n];
        for (int i = 0; i < n; i++) {
            int id = sc.nextInt();
            sc.nextLine();
            String name = sc.nextLine();
            int[] times = new int[5];
            for (int j = 0; j < 5; j++) {
                times[j] = sc.nextInt();
            }
```

```java
        //sc.nextLine();
        runners[i] = new Runner(id, name, times);
    }
    for (int i = 0; i < n; i++) {
        System.out.println("Runner ID: " + runners[i].getId());
        System.out.println("Runner Name: " + runners[i].getName());
        System.out.println("Average Time: " + runners[i].getAverage());
    }
    int minAvg = runners[0].getAverage();
    int minId = runners[0].getId();
    String fastestRunner = runners[0].getName();
    for (int i = 1; i < n; i++) {
        int currentAvg = runners[i].getAverage();
        int currentId = runners[i].getId();
        if (currentAvg < minAvg || (currentAvg == minAvg && currentId < minId)) {
            minAvg = currentAvg;
            minId = currentId;
            fastestRunner = runners[i].getName();
        }
    }
    System.out.println("Fastest Runner: " + fastestRunner + " with " + minAvg + "
minutes");
    }
}
```

Status : Correct   *Marks : 10/10*

4. 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

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

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

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

    public void setBalance(double balance) {
        this.balance = balance;
    }

    public int getId() {
        return id;
    }

    public String getName() {
        return name;
    }

    public double getBalance() {
        return balance;
    }
}
```

```java
    public void recharge(double amount) {
        if (amount >= 0) {
            this.balance += amount;
        }
    }


    public void useData(double amount) {
        if (amount <= this.balance) {
            this.balance -= amount;
        }

    }
}

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

        int N = sc.nextInt();
        sc.nextLine();

        Customer[] customers = new Customer[N];

        for (int i = 0; i < N; i++) {
            int id = sc.nextInt();
            sc.nextLine();

            String name = sc.nextLine();
            double initialBalance = sc.nextDouble();
            double rechargeAmount = sc.nextDouble();
            double usageAmount = sc.nextDouble();

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

            customers[i].recharge(rechargeAmount);


            customers[i].useData(usageAmount);
        }
```

```
    for (int i = 0; i < N; i++) {
        System.out.println("Customer ID: " + customers[i].getId());
        System.out.println("Customer Name: " + customers[i].getName());
        System.out.printf("Final Data Balance: %.1f GB\n",
customers[i].getBalance());
    }
  }
}
```

Status : Correct   *Marks : 10/10*