

Rajalakshmi Engineering College

Name: Reshma shri.s

Email: 241001194@rajalakshmi.edu.in

Roll no: 241001194

Phone: 8144945959

Branch: REC

Department: IT - Section 2

Batch: 2028

Degree: B.E - IT

Scan to verify results



2024_28_III_OOPS Using Java Lab

REC_2028_OOPS using Java_Week 6_MCQ

Attempt : 1

Total Mark : 15

Marks Obtained : 15

Section 1 : MCQ

1. Which of the following is true about method overriding in Java?

Answer

The method must have the same name, same parameters, and must be in different classes with an inheritance relationship

Status : Correct

Marks : 1/1

2. What will be the output of the following program?

```
class Vehicle {  
    String type = "Vehicle";  
}
```

```
class Car extends Vehicle {
```

```
        String type = "Car";
    }

class Test {
    public static void main(String[] args) {
        Car c = new Car();
        System.out.println(c.type);
    }
}
```

Answer

Car

Status : Correct

Marks : 1/1

3. What will be the output of the following Java program?

```
class A {
    int value = 10;
    void display() {
        System.out.println("A's display: " + value);
    }
}
class B extends A {
    int value = 20;
    void display() {
        System.out.println("B's display: " + value);
    }
}
class Test {
    public static void main(String[] args) {
        A obj = new B();
        obj.display();
        System.out.println("Value: " + obj.value);
    }
}
```

Answer

B's display: 20 Value: 10

Status : Correct

Marks : 1/1

4. What will be the output of the following program?

```
class A {  
    public int i;  
    private int j;  
}  
class B extends A {  
    void display() {  
        super.j = super.i + 1;  
        System.out.println(super.i + " " + super.j);  
    }  
}  
class inheritance {  
    public static void main(String args[]) {  
        B obj = new B();  
        obj.i=1;  
        obj.j=2;  
        obj.display();  
    }  
}
```

Answer

Compile Time Error

Status : Correct

Marks : 1/1

5. What will be the output of the following code?

```
class A {  
    int sum(int x) {  
        return x + 2;  
    }  
}
```

```
class B extends A {  
    int sum(int x) {
```

```
        return super.sum(x) * 2;
    }

class C extends B {
    int sum(int x) {
        return super.sum(x) - 3;
    }
}

class Test {
    public static void main(String[] args) {
        C obj = new C();
        System.out.println(obj.sum(4));
    }
}
```

Answer

9

Status : Correct

Marks : 1/1

6. What will be the output of the following code?

```
class A {
    void display() {
        System.out.println("Display A");
    }
}
```

```
class B extends A {
    void display() {
        System.out.println("Display B");
    }
}
```

```
class C extends B {
    void display() {
        super.display();
    }
}
```

```
        }  
    }  
  
class Test {  
    public static void main(String[] args) {  
        C obj = new C();  
        obj.display();  
    }  
}
```

Answer

Display B

Status : Correct

Marks : 1/1

7. Which of the following is the correct way for class B to inherit from class A?

Answer

class B extends A {}

Status : Correct

Marks : 1/1

8. What will be the output of the following Java program?

```
class Test {  
    void display(int a, int b) {  
        System.out.println("Method 1");  
    }  
    void display(double a, double b) {  
        System.out.println("Method 2");  
    }  
    public static void main(String[] args) {  
        Test obj = new Test();  
        obj.display(10, 10.0);  
    }  
}
```

Answer

Method 2

Status : Correct

Marks : 1/1

9. What will be the output of the following Java program?

```
class Vehicle {  
    void startEngine() {  
        System.out.println("Vehicle engine started");  
    }  
}
```

```
class Car extends Vehicle {  
    void startEngine() {  
        System.out.println("Car engine started");  
    }  
}
```

```
class Main {  
    public static void main(String[] args) {  
        Vehicle myVehicle = new Car();  
        myVehicle.startEngine();  
    }  
}
```

Answer

Car engine started

Status : Correct

Marks : 1/1

10. Select the correct keyword for implementing inheritance through the class.

Answer

extends

Status : Correct

Marks : 1/1

11. What will be the output of the following Java program?

```
class Test {  
    void show(int a) {  
        System.out.println("Integer method");  
    }  
    void show(String s) {  
        System.out.println("String method");  
    }  
    public static void main(String[] args) {  
        Test obj = new Test();  
        obj.show(null);  
    }  
}
```

Answer

String method

Status : Correct

Marks : 1/1

12. What will be the output of the following Java program?

```
class Parent {  
    void show() {  
        System.out.println("Parent class");  
    }  
}  
class Child extends Parent {  
    void show() {  
        System.out.println("Child class");  
    }  
}  
class Test {  
    public static void main(String[] args) {  
        Parent obj = new Child();  
        obj.show();  
    }  
}
```

Answer

Child class

Status : Correct

Marks : 1/1

13. What will be the output of the following program?

```
class A {  
    int x = 10;  
}  
  
class B extends A {  
    int x = 20;  
}  
  
class C extends B {  
    int x = 30;  
  
    void display() {  
        System.out.println(x);  
        System.out.println(super.x);  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        C obj = new C();  
        obj.display();  
    }  
}
```

Answer

3020

Status : Correct

Marks : 1/1

14. What will be the output of the following Java program?

```
class Vehicle {  
    void start() {  
        System.out.println("Vehicle starts");  
    }  
}  
class Car extends Vehicle {  
  
    void start() {  
        System.out.println("Car starts");  
    }  
}  
class ElectricCar extends Car {  
    void start() {  
        System.out.println("Electric Car starts silently");  
    }  
}  
class Test {  
    public static void main(String[] args) {  
        Vehicle v = new ElectricCar();  
        v.start();  
    }  
}
```

Answer

Electric Car starts silently

Status : Correct

Marks : 1/1

15. What will be the output of the following Java program?

```
class A {  
    void display() {  
        System.out.println("Class A");  
    }  
}
```

```
class B extends A {  
    void show() {  
        System.out.println("Class B");  
    }  
}
```

```
        }  
    }  
  
class C extends B {  
    void print() {  
        System.out.println("Class C");  
    }  
}
```

```
class Test {  
    public static void main(String[] args) {  
        C obj = new C();  
        obj.display();  
        obj.show();  
        obj.print();  
    }  
}
```

Answer

Class A Class B Class C

Status : Correct

Marks : 1/1

Rajalakshmi Engineering College

Name: Reshma shri.s

Email: 241001194@rajalakshmi.edu.in

Roll no: 241001194

Phone: 8144945959

Branch: REC

Department: IT - Section 2

Batch: 2028

Degree: B.E - IT

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 6_Q1

Attempt : 1

Total Mark : 10

Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Elsa subscribes to a premium service with a base monthly cost, a service tax and an extra feature cost. Assist her in writing an inheritance program that takes input for these values and calculates the total monthly cost.

Refer to the below class diagram:

Input Format

The first line of input consists of a double value, representing the base monthly cost.

The second line consists of a double value, representing the service tax.

The third line consists of a double value, representing the extra feature cost.

Output Format

The output prints "Rs. X" where X is a double value, rounded off to two decimal places.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 10.0

2.5

5.0

Output: Rs. 17.50

Answer

```
import java.util.Scanner;

// You are using Java
class subscription
{
    double baseMonthlyCost;
    double serviceTax;

    subscription(double baseMonthlyCost, double serviceTax)
    {
        this.baseMonthlyCost=baseMonthlyCost;
        this.serviceTax=serviceTax;
    }

    double calculateMonthlyCost()
    {
        return baseMonthlyCost + serviceTax;
    }
}

class PremiumSubscription extends subscription
{
    double extraFeatureCost;

    PremiumSubscription(double baseMonthlyCost,double serviceTax,double extraFeatureCost)
    {
```

```
super(baseMonthlyCost, serviceTax);
this.extraFeatureCost=extraFeatureCost;
}
double calculateMonthlyCost()
{
    return super.calculateMonthlyCost() + extraFeatureCost;
}
}

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

        double baseMonthlyCost = scanner.nextDouble();
        double serviceTax = scanner.nextDouble();
        double extraFeatureCost = scanner.nextDouble();

        PremiumSubscription premiumSubscription = new
PremiumSubscription(baseMonthlyCost, serviceTax, extraFeatureCost);

        double totalMonthlyCost = premiumSubscription.calculateMonthlyCost();

        System.out.printf("Rs. %.2f%n", totalMonthlyCost);

        scanner.close();
    }
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: Reshma shri.s

Email: 241001194@rajalakshmi.edu.in

Roll no: 241001194

Phone: 8144945959

Branch: REC

Department: IT - Section 2

Batch: 2028

Degree: B.E - IT

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 2_Q6

Attempt : 1

Total Mark : 10

Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Maya, a student in an arts and crafts class, wants to create a pattern using stars (*) in a specific format. She plans to use a program to help her construct the pattern.

Write a program that takes an integer as input and constructs the following pattern using nested for loops.

Input: 5

Output:

*

**

```
***  
****  
*****  
****  
***  
**  
*
```

Input Format

The input consists of a number (integer) representing the number of rows.

Output Format

The output displays the required pattern.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 5

Output: *

```
**  
***  
****  
*****  
***  
**  
*
```

Answer

```
// You are using Java  
import java.io.*;  
import java.util.*;  
public class Main  
{
```

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

    int mid=sc.nextInt();
    int tr=(mid*2)-1;
    int col=1;

    for(int rows=1;rows<=tr;rows++)
    {
        for(int c=1;c<=col;c++)
        {
            System.out.println("* ");
        }
        if(rows<mid)
        {
            col++;
        }
        else
        {
            col--;
        }
        System.out.println();
    }
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: Reshma shri.s

Email: 241001194@rajalakshmi.edu.in

Roll no: 241001194

Phone: 8144945959

Branch: REC

Department: IT - Section 2

Batch: 2028

Degree: B.E - IT

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 6_Q3

Attempt : 1

Total Mark : 10

Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Preethi is working on a project to automate sales tax calculations for items in a store. She wants to create a program that takes the price of an item and the sales tax rate as input and calculates the final price of the item after applying the sales tax.

Write a program using the class SalesTaxCalculator, which contains an overloaded method named calculateFinalPrice to handle both integer and double inputs. The program should also include a Main class that takes user input, calls the appropriate method from SalesTaxCalculator, and prints the final price of the item.

Formula Used: Final price = price + ((price * sales tax rate) / 100)

Input Format

The first line of input consists of an integer price (the price of the item for integer inputs).

The second line of input consists of an integer taxRate (the sales tax rate for integer inputs).

The third line of input consists of a double price (the price of the item for double inputs).

The fourth line of input consists of a double taxRate (the sales tax rate for double inputs).

Output Format

The first line of output prints an integer, representing the final price of the item after applying the sales tax for integer inputs (a and b).

The second line prints a double value, representing the final price of the item after applying the sales tax for double-value inputs (m and n), rounded to two decimal places.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 100

10

100.0

5.0

Output: 110

105.00

Answer

```
import java.util.Scanner;  
  
// You are using Java  
class SalesTaxCalculator  
{  
    int intPrice;  
    int intTaxRate;  
    double doublePrice;
```

```

        double doubleTaxRate;

        SalesTaxCalculator(int intPrice,int intTaxRate, double doubleintPrice, double
doubleTaxRate)
{
    this.intPrice=intPrice;
    this.intTaxRate=intTaxRate;
    this.doublePrice=doublePrice;
    this.doubleTaxRate=doubleTaxRate;
}
public static int calculateFinalPrice(int intPrice, int intTaxRate)
{
    return intPrice + ((intPrice * intTaxRate)/100);
}
public static double calculateFinalPrice(double doublePrice, double
doubleTaxRate)
{
    return doublePrice + ((doublePrice * doubleTaxRate)/100);
}

}

class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int intPrice = scanner.nextInt();
        int intTaxRate = scanner.nextInt();
        double doublePrice = scanner.nextDouble();
        double doubleTaxRate = scanner.nextDouble();

        int finalPriceInt = SalesTaxCalculator.calculateFinalPrice(intPrice,
intTaxRate);
        double finalPriceDouble =
SalesTaxCalculator.calculateFinalPrice(doublePrice, doubleTaxRate);

        System.out.println(finalPriceInt);
        System.out.format("%.2f", finalPriceDouble);
    }
}

```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: Reshma shri.s

Email: 241001194@rajalakshmi.edu.in

Roll no: 241001194

Phone: 8144945959

Branch: REC

Department: IT - Section 2

Batch: 2028

Degree: B.E - IT

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 6_Q4

Attempt : 1

Total Mark : 10

Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Mr.Kapoor wants to create a program to calculate the volume of a Cuboid and a Cube using method overriding.

Implements a base class Cuboid with attributes for length, width, and height. Include a method calculateVolume() that computes the volume of the cuboid.

Extends the base class with a subclass Cube representing a cube, where all sides are equal. Override the calculateVolume() method in the Cube class to compute the volume of the cube.

The program should take user input for the dimensions of the cuboid and the side length of the cube and display the calculated volumes with two decimal places.

Input Format

The first line of input consists of 3 space-separated double values, representing the cuboid length, width, and height, respectively.

The second line consists of a double value, representing the side length of the cube.

Output Format

The first line of output prints the volume of the cuboid, rounded off to two decimal places.

The second line prints the volume of the cube, rounded off to two decimal places.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 60.0 60.0 60.0
50.0

Output: Volume of Cuboid: 216000.00
Volume of Cube: 125000.00

Answer

```
import java.util.Scanner;  
// You are using Java  
class Cuboid  
{  
    double cuboidLength;  
    double cuboidWidth;  
    double cuboidHeight;  
  
    Cuboid(double cuboidLength,double cuboidWidth,double cuboidHeight)  
    {  
        this.cuboidLength=cuboidLength;  
        this.cuboidWidth=cuboidWidth;  
        this.cuboidHeight=cuboidHeight;  
    }  
}
```

```
public double calculateVolume()
{
    return cuboidLength * cuboidWidth * cuboidHeight;
}
}
class Cube extends Cuboid
{
    double side;

    Cube(double side)
    {
        super(side,side,side);
        this.side=side;
    }
    @Override
    public double calculateVolume()
    {
        return cuboidLength * cuboidLength * cuboidLength;
    }
}

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

        double cuboidLength = scanner.nextDouble();
        double cuboidWidth = scanner.nextDouble();
        double cuboidHeight = scanner.nextDouble();

        // Regular object instantiation for Cuboid
        Cuboid cuboid = new Cuboid(cuboidLength, cuboidWidth, cuboidHeight);
        System.out.printf("Volume of Cuboid: %.2f\n", cuboid.calculateVolume());

        double cubeSide = scanner.nextDouble();

        // Upcasting - Using superclass reference for subclass object (DMD)
        Cuboid cube = new Cube(cubeSide); // Upcasting
        System.out.printf("Volume of Cube: %.2f", cube.calculateVolume()); // Calls
        Cube's method dynamically

        scanner.close();
    }
}
```

}

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: Reshma shri.s

Email: 241001194@rajalakshmi.edu.in

Roll no: 241001194

Phone: 8144945959

Branch: REC

Department: IT - Section 2

Batch: 2028

Degree: B.E - IT

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 6_Q5

Attempt : 1

Total Mark : 10

Marks Obtained : 10

Section 1 : Coding

1. Problem statement:

Tim was tasked with developing a grocery shopping app. You have a class hierarchy that includes Item, Produce, and OrganicProduce. Your goal is to calculate the total cost of a shopping list, which may contain a mix of regular produce and organic produce items. Additionally, you need to apply discounts to organic items. Apply a 10% discount on organic produce items

Class Hierarchy:

Item: Base class for all items.

Produce: Subclass of Item for regular produce items.

OrganicProduce: Subclass of Produce for organic produce items.

Input Format

The first line of input consists of an integer, 'n'.

For each 'n' item, the user will provide:

- A string 'type' representing the item type ('Regular' or 'Organic').
- A string 'name' represents the item name.
- A double 'price' represents the item price.

Output Format

The output will display the total cost of the shopping list, including discounts on organic items.

Refer to the sample output for format specifications.

Sample Test Case

Input: 1

Regular Banana 1.99

Output: 1.99

Answer

```
import java.util.Scanner;  
// You are using Java  
class Item  
{  
    String name;  
    double price;  
  
    Item(String name,double price)  
    {  
        this.name=name;  
        this.price=price;  
    }  
    double calculateCost()  
    {  
        return price;  
    }
```

```
}

class Produce extends Item
{
    Produce(String name,double price)
    {
        super(name,price);
    }
}

class OrganicProduce extends Produce
{
    OrganicProduce(String name,double price)
    {
        super(name,price);
    }

    @Override
    double calculateCost()
    {
        return super.calculateCost() * 0.90;
    }
}

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

        int n = sc.nextInt();
        sc.nextLine(); // Consume newline

        double totalCost = 0.0;

        for (int i = 0; i < n; i++) {
            String type = sc.next();
            String name = sc.next();
            double price = sc.nextDouble();

            if (type.equals("Regular")) {
                Item item = new Produce(name, price);
                totalCost += item.calculateCost();
            } else if (type.equals("Organic")) {
                Item item = new OrganicProduce(name, price);
                totalCost += item.calculateCost();
            }
        }
    }
}
```

```
        } } System.out.printf("%.2f\n", totalCost);
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: Reshma shri.s

Email: 241001194@rajalakshmi.edu.in

Roll no: 241001194

Phone: 8144945959

Branch: REC

Department: IT - Section 2

Batch: 2028

Degree: B.E - IT

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 5_PAH

Attempt : 1

Total Mark : 50

Marks Obtained : 30

Section 1 : Coding

1. Problem Statement

Anjali is working as a developer for CityFitness Gym, which wants to build a system to calculate monthly membership fees for gym members based on the type of membership and the number of personal training sessions booked.

Each member's record has:

Member ID (integer) Member Name (string) Membership Type (string: "Basic", "Premium", "Elite") Number of Personal Training Sessions (integer)

The monthly fees are:

Basic – 1000 units Premium – 1500 units Elite – 2000 units

The cost of personal training sessions is 500 units per session.

The calculation rules:

Total Amount = Membership Fee + (Number of Personal Training Sessions \times 500)
If the number of sessions is more than 5, a 10% discount is applied on the total amount.
If the member has Elite membership and the total amount exceeds 4000, an additional 5% service tax is added after discount.

Anjali has been asked to implement this system using:

A class with attributes for member details. A constructor to initialize member details. Getter and Setter methods to retrieve and update member details if required. A method to calculate the final monthly fee. Objects of the class to represent members.

Finally, display each member's details and the final monthly fee.

Input Format

The first line contains an integer N, representing the number of members.

For each member:

- Next line contains Member ID (integer)
- Next line contains Member Name (string)
- Next line contains Membership Type ("Basic", "Premium", "Elite")
- Next line contains Number of Personal Training Sessions (integer)

Output Format

For each member, print:

- Member ID: <member_id>
- Member Name: <member_name>
- Final Monthly Fee: <final_fee> (The final fee must be rounded to one decimal place)

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1

1001
Ravi Kumar
Basic
3

Output: Member ID: 1001
Member Name: Ravi Kumar
Final Monthly Fee: 2500.0

Answer

```
// You are using Java
import java.io.*;
import java.util.*;
class cityfitness
{
    int memberID;
    String memberName;
    String memberType;
    int noofsessions;

    cityfitness(int memberID, String memberName, String memberType, int
noofsessions)
    {
        this.memberID=memberID;
        this.memberName=memberName;
        this.memberType=memberType;
        this.noofsessions=noofsessions;
    }
    int getmemberID()
    {
        return memberID;
    }
    String getmemberName()
    {
        return memberName;
    }
    String getmemberType()
    {
        return memberType;
    }
    int getnoofsessions()
    {
        return noofsessions;
    }
}
```

```
}

void setmemberID(int memberID)
{
    this.memberID=memberID;
}
void setmemberName(String memberName)
{
    this.memberName=memberName;
}
void setmemberType(String memberType)
{
    this.memberType=memberType;
}
void setnoofsessions(int noofsessions)
{
    this.noofsessions=noofsessions;
}
double calculate()
{
    double ta=0.0;
    double pt=500;

    if(memberType.equals("Basic"))
    {
        ta=1000+(noofsessions * pt);
    }
    if(memberType.equals("Premium"))
    {
        ta=1500+(noofsessions * pt);
    }
    if(memberType.equals("Elite"))
    {
        ta=2000+(noofsessions * pt);
    }
    if(noofsessions>5)
    {
        ta=ta*0.90;
    }
    if(memberType.equals("Elite") && ta>4000)
    {
        ta=ta*1.05;
    }
}
```

```

        return ta;
    }

}

public class Main
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        for(int i=0;i<n;i++)
        {
            int memID=sc.nextInt();
            sc.nextLine();
            String memName=sc.nextLine();
            String memtype=sc.nextLine();
            int np=sc.nextInt();

            cityfitness gym=new cityfitness(memID,memName,memtype,np);
            double totalamount=gym.calculate();

            System.out.println("Member ID: "+gym.getmemberID());
            System.out.println("Member Name: "+gym.getmemberName());
            System.out.printf("Final Monthly Fee: %.1f%n",totalamount);
        }
    }
}

```

Status : Correct

Marks : 10/10

2. Problem Statement

Neha is working as a developer for CityMovie Theatre, which wants to build a system to calculate total ticket cost for movie-goers based on the number of tickets and type of seats booked.

Each customer's booking has:

Booking ID (integer)Customer Name (string)Number of Tickets

(integer)Seat Type (string: "Standard", "Premium", "VIP")

The ticket prices are:

Standard – 250 units per ticket
Premium – 400 units per ticket
VIP – 600 units per ticket

The calculation rules:

Total Amount = Number of Tickets × Seat Price

If a customer books more than 4 tickets, they get a 10% discount on the total amount.

If the booking is for VIP seats and the total amount exceeds 3000 units, a 5% luxury tax is added after any discount.

Neha has been asked to implement this system using:

A class with attributes for booking details. A constructor to initialize booking details. Getter and Setter methods to retrieve and update booking details if required. A method to calculate the final ticket cost. Objects of the class to represent bookings.

Finally, display each customer's details and final ticket amount.

Input Format

The first line contains an integer N, representing the number of bookings.

For each booking:

- The next line contains the Booking ID (integer).
- The next line contains the Customer Name (string).
- The next line contains Number of Tickets (integer).
- The next line contains Seat Type ("Standard", "Premium", or "VIP").

Output Format

For each booking, print:

- Booking ID: <booking_id>
- Customer Name: <customer_name>
- Final Ticket Amount: <final_amount> (rounded to one decimal place)

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1

1001

Ravi Kumar

3

Standard

Output: Booking ID: 1001

Customer Name: Ravi Kumar

Final Ticket Amount: 750.0

Answer

```
// You are using Java
import java.io.*;
import java.util.*;
class citymovie
{
    int bookingid;
    String cusname;
    int nooftickets;
    String seattype;

    citymovie(int bookingid, String cusname,int nooftickets, String seattype)
    {
        this.bookingid=bookingid;
        this.cusname=cusname;
        this.nooftickets=nooftickets;
        this.seattype=seattype;
    }
    int getbookingid()
    {
        return bookingid;
    }
    String getcusname()
    {
        return cusname;
    }
```

```
int getnooftickets()
{
    return nooftickets;
}
String getseattype()
{
    return seattype;
}
void setbookingid(int bookingid)
{
    this.bookingid=bookingid;
}
void setcusname(String cusname)
{
    this.cusname=cusname;
}
void setnooftickets(int nooftickets)
{
    this.nooftickets=nooftickets;
}
void setseattype(String seattype)
{
    this.seattype=seattype;
}
double calculate()
{
    double ta=0.0;
    if(seattype.equals("Standard"))
    {
        ta=(nooftickets*250);
    }
    if(seattype.equals("Premium"))
    {
        ta=(nooftickets*400);
    }
    if(seattype.equals("VIP"))
    {
        ta=(nooftickets*600);
    }
    if(nooftickets > 4)
    {
        ta=ta*0.90;
    }
}
```

```

        }
        if(seattype.equals("VIP")&& ta>3000)
        {
            ta=ta*1.05;
        }
        return ta;
    }
}

public class Main
{
    public static void main(String[] arg)
    {
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        for(int i=0;i<n;i++)
        {
            int bID=sc.nextInt();
            sc.nextLine();
            String cn=sc.nextLine();
            int nt=sc.nextInt();
            sc.nextLine();
            String st=sc.nextLine();

            citymovie theatre=new citymovie(bID,cn,nt,st);
            double totalamount=theatre.calculate();

            System.out.println("Booking Id: "+theatre.getbookingid());
            System.out.println("Customer Name: "+theatre.getcusname());
            System.out.printf("Final Ticket Amount: %.1f%n",totalamount);
        }
    }
}

```

Status : Correct

Marks : 10/10

3. Problem Statement

Neha is working as a developer for CityQuiz Platform, which wants to build a system to calculate quiz scores and identify top scorers among participants.

Each participant's record has:

Participant ID (integer) Participant Name (string) An array of scores in 5 quiz rounds (integers, each between 0 and 100)

The system must calculate:

Total Score = sum of scores in all 5 rounds. Average Score = Total Score ÷ 5. If a participant scores above 80 in all rounds, a bonus of 10 points is added to the total score. Identify the Top Scorer among all participants. If two participants have the same total score, the one with the lower Participant ID is considered the top scorer.

Neha has been asked to implement this system using:

A class with attributes for participant details. A constructor to initialize participant details. Getter and setter methods to retrieve or update participant details. A method to calculate total score and average score (including bonus if applicable). Objects of the class to represent participants.

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

Input Format

The first line of input contains an integer N, representing the number of participants.

For each participant:

- Next line: Participant ID (integer)
- Next line: Participant Name (string)
- Next line: 5 integers separated by spaces (scores for 5 quiz rounds)

Output Format

For each participant:

- Participant ID: <participant_id>
- Participant Name: <participant_name>
- Total Score: <total_score>
- Average Score: <average_score>

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

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1

1001

Ravi Kumar

85 90 88 92 87

Output: Participant ID: 1001

Participant Name: Ravi Kumar

Total Score: 452

Average Score: 90

Top Scorer: Ravi Kumar with 452 points

Answer

```
// You are using Java
import java.io.*;
import java.util.*;
class cityquiz
{
    int parid;
    String parname;
    int[] scores;
    int totalscore;
    double avgscore;

    cityquiz(int parid,String parname,int[] scores)
    {
        this.parid=parid;
        this.parname=parname;
        this.scores=scores;
        this.calculate();
    }
    void calculate()
    {
```

```
int sum=0;
boolean bonusele=true;

for(int score : scores)
{
    sum+=score;
    if(score<=80)
    {
        bonusele=false;
    }
}
this.totalscore=sum;

if(bonusele)
{
    this.totalscore += 10;
}
this.avgscore=(double) this.totalscore/5.0;
}

int getparid()
{
    return parid;
}

String getparname()
{
    return parname;
}

int gettotalscore()
{
    return totalscore;
}

double getavgscore()
{
    return avgscore;
}

}

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

```

int n=sc.nextInt();
cityquiz topscorer=null;

for(int i=0;i<n;i++)
{
    int pID=sc.nextInt();
    sc.nextLine();
    String pn=sc.nextLine();
    int scores[]=new int[5];
    for(int j=0;j<5;j++)
    {
        scores[j]=sc.nextInt();
    }
    System.out.println("Top Scorer: "+topscorer.getparname() + " with
"+topscorer.gettotalscore() + " points");
    cityquiz platform = new cityquiz(pID,pn,scores);

    System.out.println("Participant ID: "+platform.getparid());
    System.out.println("Participant Name: "+platform.getparname());
    System.out.println("Total Score: "+platform.gettotalscore());
    System.out.printf("Average Score: %.0fxn",platform.getavgscore());

    if(topscorer==null || platform.gettotalscore()>topscorer.gettotalscore() ||
(platform.gettotalscore()==topscorer.gettotalscore() &&
platform.getparid()<topscorer.getparid()))
    {
        topscorer=platform;
    }
}
}

```

Status : Skipped

Marks : 0/10

4. Problem Statement

Ravi is working as a developer for SecureLogin Systems, which wants to build a system to evaluate the strength of user passwords.

Each user record has:

User ID (integer)User Name (string)Password (string)

The system must calculate whether a password is strong or weak.

A password is considered strong if it meets all of the following conditions:

At least 8 characters long.Contains at least one uppercase letter.Contains at least one lowercase letter.Contains at least one digit.Contains at least one special character (from !@#\$%^&*).

Ravi has been asked to implement this system using:

A class with attributes for user details.A constructor to initialize user details.Getter and setter methods to retrieve or update user details.A method to check whether the password is strong.Objects of the class to represent users.

Finally, display each user's details and indicate whether their password is Strong or Weak.

Input Format

The first line contains an integer N, representing the number of users.

For each user:

The next line contains the User ID (integer).

The next line contains the User Name (string).

The next line contains the Password (string).

Output Format

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

User ID: <user_id>

User Name: <user_name>

Password: <password>

Password Strength: <Strong/Weak>

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1

1001

Ravi Kumar

Abc@1234

Output: User ID: 1001

User Name: Ravi Kumar

Password: Abc@1234

Password Strength: Strong

Answer

```
// You are using Java
import java.io.*;
import java.util.*;
class securelogin
{
    int userid;
    String username;
    String password;

    securelogin(int userid,String username,String password)
    {
        this.userid=userid;
        this.username=username;
        this.password=password;
    }
    int getuserid()
    {
        return userid;
    }
    String getusername()
    {
        return username;
    }
    String getpassword()
    {
```

```
        return password;
    }
    String calculate()
    {
        boolean hasUppercase=false;
        boolean hasLowercase=false;
        boolean hasDigit=false;
        boolean hasSpecialChar=false;
        if(password.length()<8)
        {
            return "weak";
        }
        for(char ch : password.toCharArray())
        {
            if(Character.isUpperCase(ch))
            {
                hasUppercase=true;
            }
            else if(Character.isLowerCase(ch))
            {
                hasLowercase=true;
            }
            else if(Character.isDigit(ch))
            {
                hasDigit = true;
            }
            else if("!@#$%^&*".indexOf(ch)>=0)
            {
                hasSpecialChar=true;
            }
        }
        if(hasUppercase && hasLowercase)
        {
            return "Strong";
        }
        else
        {
            return "weak";
        }
    }
}
```

```

public class Main
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        for(int i=0;i<n;i++)
        {
            int userid=sc.nextInt();
            sc.nextLine();
            String username=sc.nextLine();
            String password=sc.nextLine();

            securelogin system=new securelogin(userid,username,password);
            String strength=system.calculate();

            System.out.println("User ID: "+system.getuserid());
            System.out.println("User Name: "+system.getusername());
            System.out.println("Password: "+system.getpassword());
            System.out.println("Password Strength: "+strength);

        }
    }
}

```

Status : Correct

Marks : 10/10

5. Problem Statement

Each customer at the bank has an Account Number, Customer Name, and an Initial Balance. The bank allows two types of transactions:

Deposit – Increases the balance. Withdrawal – Decreases the balance, but only if enough funds are available. If the withdrawal amount exceeds the available balance, the transaction should be skipped, and the balance should remain unchanged.

You are required to implement this banking system by:

Creating a class with the necessary attributes to store account details.

Using a constructor to initialize the account details when a new account is created. Providing setter methods to update the details if required. Providing getter methods to retrieve account details. Creating objects of this class to represent different customers, where each customer can perform deposits and withdrawals.

Instructions:

Implement the class to store account details. Implement the logic for performing deposit and withdrawal transactions. Ensure that withdrawals don't exceed the available balance. After performing the transactions, print the account number, customer name, and final balance.

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

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

class Account {
    private int accountNumber;
    private String customerName;
    private double balance;

    public Account(int accountNumber, String customerName, double balance) {
        this.accountNumber = accountNumber;
        this.customerName = customerName;
        this.balance = balance;
    }

    public void setAccountNumber(int accountNumber) {
        this.accountNumber = accountNumber;
    }

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

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

    public int getAccountNumber() {
        return accountNumber;
    }

    public String getCustomerName() {
        return customerName;
    }

    public double getBalance() {
```

```
        return balance;
    }

    public void deposit(double amount) {
        if (amount >= 0) balance += amount;
    }

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

class CityBankApp {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        sc.nextLine();
        for (int i = 0; i < n; i++) {
            int accNo = Integer.parseInt(sc.nextLine());
            sc.nextLine();
            String name = sc.nextLine();
            double initBal = sc.nextDouble();

            double depositAmount = sc.nextDouble();

            double withdrawAmount = sc.nextDouble();

            Account acc = new Account(accNo, name, initBal);
            acc.deposit(depositAmount);
            acc.withdraw(withdrawAmount);

            System.out.println("Account Number: " + acc.getAccountNumber());
            System.out.println("Customer Name: " + acc.getCustomerName());
            System.out.printf("Final Balance: %.1f%n", acc.getBalance());
        }
        sc.close();
    }
}
```

Status : Skipped

Marks : 0/10

Rajalakshmi Engineering College

Name: Reshma shri.s

Email: 241001194@rajalakshmi.edu.in

Roll no: 241001194

Phone: 8144945959

Branch: REC

Department: IT - Section 2

Batch: 2028

Degree: B.E - IT

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

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.io.*;
import java.util.*;

class player
{
    int playerid;
    String playername;
    int[] points;
    int totalscore;

    public player(int playerid, String playername, int[] points)
    {
        this.playerid = playerid;
        this.playername = playername;
        this.points = points;
        calculate();
    }

    int getplayerid()
    {
        return playerid;
    }
    String getplayername()
    {
        return playername;
    }
    int gettotalscore()
    {
        return totalscore;
    }
    void setplayerid(int playerid)
    {
        this.playerid = playerid;
    }
    void setplayername(String playername)
```

```
        this.playername=playername;
    }
    void setpoints(int[] points)
    {
        this.points=points;
        calculate();
    }
    void calculate()
    {
        totalscore=0;
        for(int p: points)
        {
            totalscore+=p;
        }
    }
}
public class Main
{
    public static void main(String[] arg)
    {
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        sc.nextLine();
        player[] players=new player[n];
        for(int i=0;i<n;i++)
        {
            int id=Integer.parseInt(sc.nextLine().trim());
            String name=sc.nextLine().trim();
            int[] points=new int[5];
            String[] scores=sc.nextLine().split(" ");
            for(int j=0;j<5;j++)
            {
                points[j]=Integer.parseInt(scores[j]);
            }
            players[i]=new player(id,name,points);
        }
        for(player p : players)
        {
            System.out.println("Player ID: "+p.getplayerid());
```

```
        System.out.println("Player Name: "+p.getplayername());
        System.out.println("Total Score: "+p.gettotalscore());
    }
    player topscorer=players[0];
    for(int i=1;i<n;i++)
    {
        if(players[i].gettotalscore()>topscorer.gettotalscore() ||
(players[i].gettotalscore()==topscorer.gettotalscore() &&
players[i].getplayerid()<topscorer.getplayerid()))
        {
            topscorer=players[i];
        }
    }
    System.out.println("Top scorer: "+
topscorer.getplayername()+"with"+topscorer.gettotalscore()+"points");
}
}
```

Status : Correct

Marks : 10/10

2. 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;
import java.util.ArrayList;
class Runner
{
    int runnerid;
    String runnername;
    int[] times;
    double avgtime;

    Runner(int runnerid, String runnername, int[] times)
    {
        this.runnerid = runnerid;
        this.runnername = runnername;
        this.times = times;
        calculate();
    }
    void calculate()
    {
        int sum = 0;
        for (int time : times)
        {
            sum += time;
        }
        this.avgtime = (double) sum / times.length;
    }
    int getrunnerid()
    {
        return runnerid;
    }
    String getrunnername()
    {
        return runnername;
    }
    double getavgtime()
    {
        return avgtime;
    }
}
```

```
}

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

        ArrayList<Runner> runners=new ArrayList<>();

        for(int i=0;i<n;i++)
        {
            int runnerid=sc.nextInt();
            sc.nextLine();
            String runnername=sc.nextLine();
            int[] times=new int[5];
            for(int j=0;j<5;j++)
            {
                times[j]=sc.nextInt();
            }
            runners.add(new Runner(runnerid,runnername,times));
        }
        Runner fastestrunner=null;

        for(Runner runner:runners)
        {
            System.out.println("Runner ID: "+runner.getrunnerid());
            System.out.println("Runner Name: "+runner.getrunnername());
            System.out.println("Average Time: "+(int)runner.getavgtime());

            if(fastestrunner == null || runner.getavgtime() < fastestrunner.getavgtime()
            || (runner.getavgtime() == fastestrunner.getavgtime() && runner.getrunnerid() <
            fastestrunner.getrunnerid()))
            {
                fastestrunner=runner;
            }
        }
        if(fastestrunner != null)
        {
            System.out.println("Fastest Runner: "+ fastestrunner.getrunnername() +
```

```
        "with" +(int)fastestrunner.getavgtime()+"minutes");
    }
}
}
```

Status : Correct

Marks : 10/10

3. 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 liter
For liters above 1000 5 per liter
If 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
```

```
{
```

```
    int customerid;
```

```
    String customername;
```

```
    double litersconsumed;
```

```
    double finalbill;
```

```
    customer(int customerid,String customername,double litersconsumed)
```

```
{
```

```
    this.customerid=customerid;
```

```
this.customername=customername;
this.litersconsumed=litersconsumed;
calculate();
}
void setcustomerid(int customerid)
{
    this.customerid=customerid;
}
void setcustomername(String customername)
{
    this.customername=customername;
}
void settitersconsumed(double litersconsumed)
{
    this.litersconsumed=litersconsumed;
    calculate();
}
int getcustomerid()
{
    return customerid;
}
String getcustomername()
{
    return customername;
}
double getfinalbill()
{
    return finalbill;
}
void calculate()
{
    double bill=0;
    double consumed=litersconsumed;
    if(consumed<=500)
    {
        bill=consumed*2;
    }
    else if(consumed<=1000)
    {
        bill=(500*2)+((consumed-500)*3);
    }
    else
```

```

        {
            bill=(500*2)+(500*3)+((consumed-1000)*5);
        }
        if(bill>3000)
        {
            bill *= 0.90;
        }
        this.finalbill=bill;
    }
}
public class Main
{
    public static void main(String[] arg)
    {
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        for(int i=0;i<n;i++)
        {
            int customerid=sc.nextInt();
            sc.nextLine();
            String customername=sc.nextLine();
            double litresconsumed=sc.nextDouble();

            customer cus=new
customer(customerid,customername,litresconsumed);
            System.out.println("Customer ID: "+cus.getcustomerid());
            System.out.println("Customer Name: "+cus.getcustomername());
            System.out.printf("Final Bill: %.1f%n",cus.getfinalbill());
        }
    }
}

```

Status : Correct

Marks : 10/10

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 customer
{
    int customerid;
    String customername;
    double unitsconsumed;
    double finalbill;

    customer(int customerid, String customername, double unitsconsumed)
    {
        this.customerid=customerid;
        this.customername=customername;
        this.unitsconsumed=unitsconsumed;
        calculate();
    }
    void setcustomerid(int customerid)
    {
        this.customerid=customerid;
    }
    void setcustomername(String customername)
    {
        this.customername=customername;
    }
    void setunitsconsumed(double unitsconsumed)
    {
        this.unitsconsumed=unitsconsumed;
        calculate();
    }
    int getcustomerid()
    {
        return customerid;
    }
}
```

```
String getcustomername()
{
    return customername;
}
double getfinalbill()
{
    return finalbill;
}
void calculate()
{
    double bill=0;
    double consumed=unitsconsumed;

    if(consumed<=50)
    {
        bill=consumed*4;
    }
    else if(consumed<=150)
    {
        bill=(50*4)+((consumed-50)*6);
    }
    else
    {
        bill=(50*4)+(100*6)+((consumed-150)*8);
    }
    if(bill>2000)
    {
        bill*=0.85;
    }
    this.finalbill=bill;
}

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

        for(int i=0;i<n;i++)
        {
```

```
int customerid=sc.nextInt();
sc.nextLine();
String customername=sc.nextLine();
double unitsconsumed=sc.nextDouble();

customer cus=new
customer(customerid,customername,unitsconsumed);
System.out.println("Customer ID: "+ cus.getcustomerid());
System.out.println("Customer Name: "+ cus.getcustomername());
System.out.printf("Final Bill: %.1f%n", cus.getfinalbill());
}
```

Status : Correct

Marks : 10/10