

Rajalakshmi Engineering College

Name: Bala murugan
Email: 241901014@rajalakshmi.edu.in
Roll no: 241901014
Phone: 9962293932
Branch: REC
Department: CSE (CS) - Section 1
Batch: 2028
Degree: B.E - CSE (CS)

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. 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

2. 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

3. 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

4. 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

5. 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

6. 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

7. 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

8. 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

9. 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

10. 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

11. 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

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

Answer

extends

Status : Correct

Marks : 1/1

13. 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

14. 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

15. 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

Rajalakshmi Engineering College

Name: Bala murugan
Email: 241901014@rajalakshmi.edu.in
Roll no: 241901014
Phone: 9962293932
Branch: REC
Department: CSE (CS) - Section 1
Batch: 2028
Degree: B.E - CSE (CS)

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;
```

```
    double extraFeatureCost;
```

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

```
        this.baseMonthlyCost = baseMonthlyCost;
```

```
        this.serviceTax = serviceTax;
```

```
        this.extraFeatureCost = extraFeatureCost;
```

```
    }
```

```
}
```

```
class PremiumSubscription extends Subscription{
```

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

```
        super(baseMonthlyCost,serviceTax,extraFeatureCost);
```

```
    }
```

```
    public double calculateMonthlyCost(){
```

```
        return baseMonthlyCost+serviceTax+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: Bala murugan
Email: 241901014@rajalakshmi.edu.in
Roll no: 241901014
Phone: 9962293932
Branch: REC
Department: CSE (CS) - Section 1
Batch: 2028
Degree: B.E - CSE (CS)

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 6_Q2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Alice is managing an online store and wants to implement a program using inheritance to calculate the selling price of products after applying discounts.

Guide her by following the instructions:

Create a base class called Product with a public double attribute price. Create a subclass called DiscountedProduct, which extends Product and includes a private double attribute discount rate. This subclass has a method called calculateSellingPrice() to determine the final selling price after applying the discount.

Formula: Discounted selling price = price * (1 - discount rate)

Input Format

The first line of input consists of a double value p, the initial price of the product.

The second line consists of a double value d, the discount rate.

Output Format

The output prints "Rs. X", where X is a double value, representing the calculated discounted selling price, rounded off to two decimal places.

If the discount rate is greater than 1, print "Not applicable".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 50.00

0.20

Output: Rs. 40.00

Answer

```
import java.util.Scanner;
```

```
// You are using Java
```

```
class Product{
```

```
    double initialPrice;
```

```
    double discountRate;
```

```
    Product(double initialPrice,double discountRate){
```

```
        this.initialPrice = initialPrice;
```

```
        this.discountRate = discountRate;
```

```
    }
```

```
}
```

```
class DiscountedProduct extends Product{
```

```
    DiscountedProduct(double initialPrice,double discountRate){
```

```
        super(initialPrice,discountRate);
```

```
    }
```

```
    public double calculateSellingPrice(){
```

```
        return initialPrice*(1-discountRate);
```

```
    }
```

```
}
```

```
class ProductPricing {
```

```
public static void main(String[] args) {  
    Scanner scanner = new Scanner(System.in);  
  
    double initialPrice = scanner.nextDouble();  
    double discountRate = scanner.nextDouble();  
    DiscountedProduct discountedProduct = new  
DiscountedProduct(initialPrice, discountRate);  
    double sellingPrice = discountedProduct.calculateSellingPrice();  
  
    if (sellingPrice >= 0) {  
        System.out.printf("Rs. %.2f%n", sellingPrice);  
    } else {  
        System.out.println("Not applicable");  
    }  
    scanner.close();  
}  
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: Bala murugan
Email: 241901014@rajalakshmi.edu.in
Roll no: 241901014
Phone: 9962293932
Branch: REC
Department: CSE (CS) - Section 1
Batch: 2028
Degree: B.E - CSE (CS)

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{
```

```
    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: Bala murugan
Email: 241901014@rajalakshmi.edu.in
Roll no: 241901014
Phone: 9962293932
Branch: REC
Department: CSE (CS) - Section 1
Batch: 2028
Degree: B.E - CSE (CS)

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 cubeSide;
        Cube(double cubeSide){
            super(cubeSide,cubeSide,cubeSide);
        }
    }

    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: Bala murugan
Email: 241901014@rajalakshmi.edu.in
Roll no: 241901014
Phone: 9962293932
Branch: REC
Department: CSE (CS) - Section 1
Batch: 2028
Degree: B.E - CSE (CS)

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{
    public String name;
    public double price;
    public Item(String name,double price){
        this.name = name;
        this.price = price;
    }
    public double calculateCost(){
        return price;
    }
}

class Produce extends Item{
    public Produce(String name,double price){
```



```

        super(name,price);
    }
}

class OrganicProduce extends Produce{
    public OrganicProduce(String name,double price){
        super(name,price);
    }
    @Override
    public double calculateCost(){
        return price*0.9;
    }
}

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: Bala murugan
Email: 241901014@rajalakshmi.edu.in
Roll no: 241901014
Phone: 9962293932
Branch: REC
Department: CSE (CS) - Section 1
Batch: 2028
Degree: B.E - CSE (CS)

Scan to verify results



2024_28_III_OOPS Using Java Lab

REC_2028_OOPS using Java_Week 6_PAH

Attempt : 1
Total Mark : 40
Marks Obtained : 40

Section 1 : Coding

1. Problem Statement

Ram is designing a program to calculate the Body Mass Index (BMI). Your task is to assist him by following the given specifications.

Create a base class BMIcalculator with a method calculateBMI() to compute BMI using the formula $\text{weight} / (\text{height} * \text{height})$.

Extend the class with a subclass CustomBMIcalculator that overrides the method calculateBMI() to calculate BMI based on custom criteria, assigning categories such as "Underweight," "Normal Weight," "Overweight," or "Obese."

BMI < 18.5, category = "Underweight" BMI >= 18.5 & < 24.9, category = "Normal Weight" BMI >= 25 & < 29.9, category = "Overweight" else category = "Obese"

Implement user input for weight and height and display both the standard and custom BMI calculations.

Input Format

The first line of input consists of a double value, representing the weight in kgs.

The second line consists of a double value, representing the height in meters.

Output Format

The first line of output prints: "Standard BMI Calculation:"

The second line of output prints: "BMI: " followed by the calculated BMI value (to two decimal places).

The third line of output prints: "Custom BMI Calculation:"

The fourth line of output prints: "Category: " followed by the BMI category.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 69.7

2.6

Output: Standard BMI Calculation:

BMI: 10.31

Custom BMI Calculation:

Category: Underweight

Answer

```
import java.util.Scanner;
```

```
class BMIcalculator {  
    protected double weight; // in kilograms  
    protected double height; // in meters  
    public BMIcalculator(double weight, double height) {
```

```

        this.weight = weight;
        this.height = height;
    }

    public double calculateBMI() {
        return weight / (height * height);
    }

    public void displayBMI() {
        double bmi = calculateBMI();
        System.out.printf("BMI: %.2f\n", bmi);
    }
}

class CustomBMIcalculator extends BMIcalculator {
    private String category;

    public CustomBMIcalculator(double weight, double height) {
        super(weight, height);
    }

    public double calculateBMI() {
        double bmi = super.calculateBMI();

        if (bmi < 18.5) {
            category = "Underweight";
        } else if (bmi >= 18.5 && bmi < 24.9) {
            category = "Normal Weight";
        } else if (bmi >= 25 && bmi < 29.9) {
            category = "Overweight";
        } else {
            category = "Obese";
        }

        return bmi;
    }

    public void displayCustomBMI() {
        double bmi = calculateBMI();
        System.out.printf("Category: %s", category);
    }
}

```

```

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

        double weight = scanner.nextDouble();
        double height = scanner.nextDouble();

        BMlcalculator bmiCalculator = new BMlcalculator(weight, height);
        System.out.println("Standard BMI Calculation:");
        bmiCalculator.displayBMI();

        CustomBMlcalculator customBMlcalculator = new
        CustomBMlcalculator(weight, height);
        System.out.println("Custom BMI Calculation:");
        customBMlcalculator.displayCustomBMI();

        scanner.close();
    }
}

```

Status : Correct

Marks : 10/10

2. Problem Statement

In a company, each manager has a unique employee ID and a monthly salary. You are required to design a program that will calculate and display the annual(12 months) salary of a manager based on the input details provided by the user.

Implement the solution using a single inheritance approach.

Employee: The base class with attributes name and employeeID.

Manager: The derived class inheriting from Employee, with an additional attribute salary.

Input Format

The first line of input consists of a string name, representing the manager's name.

The second line of input consists of an integer employeeID, representing the manager's employee ID.

The third line of input consists of a double salary, representing the manager's monthly salary.

Output Format

The first line of output prints: Name: <name>

The second line of output prints: Annual Salary: Rs. <annual_salary> (rounded to two decimal places).

Refer to the sample output for formatting specifications.

Sample Test Case

Input: Davis

234

28750.75

Output: Name: Davis

Annual Salary: Rs. 345009.00

Answer

```
import java.util.Scanner;
import java.text.DecimalFormat;

// You are using Java
class Employee{
    String name;
    int employeeID;
    double salary;
    Employee(String name,int employeeID,double salary){
        this.name = name;
        this.employeeID = employeeID;
        this.salary = salary;
    }
}

class Manager extends Employee{
```

```

    Manager(String name,int employeeID,double salary){
        super(name,employeeID,salary);
    }
    public double calculateAnnualSalary(){
        return salary*12;
    }
}

class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        DecimalFormat df = new DecimalFormat("0.00");

        String name = scanner.nextLine();
        int employeeID = scanner.nextInt();
        double salary = scanner.nextDouble();

        Manager manager = new Manager(name, employeeID, salary);

        System.out.println("Name: " + manager.name);
        System.out.println("Annual Salary: Rs. " +
df.format(manager.calculateAnnualSalary()));

        scanner.close();
    }
}

```

Status : Correct

Marks : 10/10

3. Problem Statement

John is planning a long road trip and wants to calculate the distance his car can travel based on its speed and fuel capacity. As John knows that different cars have different fuel efficiencies, he wants a program that can help him estimate the travel distance for any given car.

To do this, you are tasked with creating a program that calculates the travel distance of a car based on its speed and fuel capacity. The calculation is simple and follows the formula:

$\text{Travel Distance} = \text{Speed} * \text{Fuel Capacity}$

You need to model this system using a Vehicle class and a Car class. The Vehicle class will have attributes for the speed and fuel capacity, while the Car class will inherit from the Vehicle class and include a method to calculate the travel distance.

Input Format

The first line of input consists of a double value representing the speed of the car in km/h.

The second line of input consists of a double value representing the fuel capacity of the car in liters.

Output Format

The first line should print "Speed: X km/h", where X is the speed of the car, rounded to two decimal places.

The second line should print "Fuel Capacity: Y liters", where Y is the fuel capacity of the car, rounded to two decimal places.

The third line should print "Travel Distance: Z km", where Z is the total travel distance the car can cover, rounded to two decimal places.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 10.0

1.0

Output: Speed: 10.00 km/h

Fuel Capacity: 1.00 liters

Travel Distance: 10.00 km

Answer

```
import java.util.Scanner;
```

```
class Vehicle {
```

```
    public double speed;
```



```

    public double fuelCapacity;
}

class Car extends Vehicle {
    public Car(double speed, double fuelCapacity) {
        this.speed = speed;
        this.fuelCapacity = fuelCapacity;
    }

    public double calculateTravelDistance() {
        return speed * fuelCapacity;
    }
}

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

        double speed = scanner.nextDouble();
        double fuelCapacity = scanner.nextDouble();

        Car car = new Car(speed, fuelCapacity);

        System.out.println("Speed: " + String.format("%.2f", car.speed) + " km/h");
        System.out.println("Fuel Capacity: " + String.format("%.2f", car.fuelCapacity)
+ " liters");
        System.out.println("Travel Distance: " + String.format("%.2f",
car.calculateTravelDistance()) + " km");

        scanner.close();
    }
}

```

Status : Correct

Marks : 10/10

4. Problem Statement

Sharon, a software developer, is working on a project to automate velocity calculations for various objects. She wants to create a class named VelocityCalculator with overloaded methods calculateVelocity to calculate the velocity. One method will accept distance in meters and time in

seconds as integers, while another will accept distance and time as doubles.

Help her in completing the project.

Formula: $\text{Velocity} = \text{distance} / \text{time}$

Input Format

The first line of input consists of an integer, representing the distance in meters (for the integer method).

The second line consists of an integer, representing the time in seconds (for the integer method).

The third line consists of a double value, representing the distance in meters (for the double method).

The fourth line consists of a double value, representing the time in seconds (for the double method).

Output Format

The first line prints the velocity calculated using the integer inputs in the format:

Velocity with integer inputs: <velocity> m/s

The second line prints the velocity calculated using the double inputs in the format:

Velocity with double inputs: <velocity> m/s

Note:

The velocity for the double inputs should be printed with two decimal places.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 100

10

100.5

10.2

Output: Velocity with integer inputs: 10 m/s

Velocity with double inputs: 9.85 m/s

Answer

```
import java.util.Scanner;

class VelocityCalculator {

    public static int calculateVelocity(int distance, int time) {
        return distance / time;
    }

    public static double calculateVelocity(double distance, double time) {
        return distance / time;
    }
}

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

        int distanceInt = scanner.nextInt();
        int timeInt = scanner.nextInt();

        double distanceDouble = scanner.nextDouble();
        double timeDouble = scanner.nextDouble();

        int velocityInt = VelocityCalculator.calculateVelocity(distanceInt, timeInt);
        double velocityDouble =
        VelocityCalculator.calculateVelocity(distanceDouble, timeDouble);

        System.out.println("Velocity with integer inputs: " + velocityInt + " m/s");
        System.out.printf("Velocity with double inputs: %.2f m/s", velocityDouble);

        scanner.close();
    }
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: Bala murugan
Email: 241901014@rajalakshmi.edu.in
Roll no: 241901014
Phone: 9962293932
Branch: REC
Department: CSE (CS) - Section 1
Batch: 2028
Degree: B.E - CSE (CS)

Scan to verify results



2024_28_III_OOPS Using Java Lab

REC_2028_OOPS using Java_Week 6_CY

Attempt : 1
Total Mark : 40
Marks Obtained : 40

Section 1 : Coding

1. Problem Statement

Arun wants to calculate the age gap between the grandfather and the son and determine the father's age after 5 years.

Your task is to assist him in developing a program using three classes: GrandFather, Father, and Son, where the GrandFather stores the grandfather's age, the Father extends GrandFather to include the father's age and calculates his age after 5 years, and Son extends Father to include the son's age and calculate the age difference between the grandfather and the son.

Input Format

The input consists of three integers representing the ages of the grandfather, father, and son, one per line.

Output Format

The first line of output prints "Grandfather and son's age gap:" followed by an integer representing the age gap between the grandfather and the son, ending with "years".

The second line prints "Father's Age:" followed by an integer representing the father's age after 5 years, ending with "years".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 50

30

3

Output: Grandfather and son's age gap: 47 years

Father's Age: 35 years

Answer

```
import java.util.Scanner;
```

```
// You are using Java
```

```
class Son{
```

```
    private int grandfather;
```

```
    private int father;
```

```
    private int son;
```

```
    public void setGrandfatherAge(int grandfather){
```

```
        this.grandfather = grandfather;
```

```
    }
```

```
    public void setFatherAge(int father){
```

```
        this.father = father;
```

```
    }
```

```
    public void setSonAge(int son){
```

```
        this.son = son;
```

```
    }
```

```
    public int calculateGrandfatherSonAgeDifference(){
```

```
        return this.grandfather-this.son;
```

```
    }
```

```
    public int calculateFatherAgeAfter5Years(){
```

```
        return this.father+5;
```

```

    }
}

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

        int grandfatherAge = scanner.nextInt();
        son.setGrandfatherAge(grandfatherAge);

        int fatherAge = scanner.nextInt();
        son.setFatherAge(fatherAge);

        int sonAge = scanner.nextInt();
        son.setSonAge(sonAge);

        System.out.println("Grandfather and son's age gap: "+
            son.calculateGrandfatherSonAgeDifference() + " years");

        int fatherAgeAfter5Years = son.calculateFatherAgeAfter5Years();
        System.out.println("Father's Age: " + fatherAgeAfter5Years + " years");
    }
}

```

Status : Correct

Marks : 10/10

2. Problem Statement

A painter needs to determine the cost to paint different shapes based on their surface area. The program should be designed to handle the area of a sphere and calculate the total painting cost using the following formulas:

Area of sphere: $\text{Area} = 4 * \pi * r^2$ where $\pi = 3.14$
 Total painting cost: $\text{Cost} = \text{cost per square meter} * \text{area of sphere}$

The program will consist of three classes:

Shape class: This class should set the shape type and radius.

Area class: This class should extend Shape to calculate the area.
 Cost class: This class should extend Area to calculate the total painting cost.

Input Format

The input consists of a string representing the shape type, a double value representing the radius, and another double value representing the cost per square meter on each line.

Output Format

For a valid shape type of "Sphere":

- The first line prints: "Area of Sphere is: <calculated_area>" rounded to two decimal places.
- The second line prints: "Cost to paint the shape is: <total_painting_cost>" rounded to two decimal places.

For any other shape types, print: "Invalid type".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: Sphere

3.4

5.8

Output: Area of Sphere is: 145.19

Cost to paint the shape is: 842.12

Answer

```
import java.util.Scanner;

// You are using Java
class Shape{
    public String shape;
    public double radius;

    public void setShape(String shape,Scanner scanner){
        this.shape = shape;
        this.radius = scanner.nextDouble();
    }
}

class Area extends Shape{
```



```

double area;
public void calculateArea(){
    if(shape.equals("Sphere")){
        area = 4*3.14*radius*radius;
        System.out.printf("Area of Sphere is: %.2f\n",area);
    }
    else{
        System.out.println("Invalid type");
        area = -1;
    }
}
}

class Cost extends Area{
    double cost;
    public void setCost(double cost){
        this.cost = cost;
    }
    public void calculateCost(){
        if(area != -1){
            double total = area*cost;
            System.out.printf("Cost to paint the shape is: %.2f\n",total);
        }
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String s = scanner.next();
        Cost shape = new Cost();
        shape.setShape(s, scanner);
        double costToPaint = scanner.nextDouble();
        shape.calculateArea();
        shape.setCost(costToPaint);
        shape.calculateCost();
    }
}

```

Status : Correct

Marks : 10/10

3. Problem Statement

Adams has a reputation company with a great number of employees. He must calculate the salary weekly according to the hourly rate and working hours. Create a program to define a class Employee with attributes name and hourly rate. Create a subclass HourlyEmployee that calculates the weekly salary based on the number of hours worked.

(The first 40 hours are based on the regular hour rate. If the work hours are greater than 40 then the work wage is 1.5 times the hourly rate)

Note: Use Math(Math.max, Math.min) functions .

Example

Input:

Chris

10

45

Output:

Weekly Salary: Rs.475.00

Explanation:

Calculation:

The first 40 hours are paid normally: $40 \times 10 = 400.00$ The extra 5 hours are paid at 1.5 times the hourly rate: $5 \times (10 \times 1.5) = 5 \times 15 = 75.00$ Total salary: $400.00 + 75.00 = 475.00$

Input Format

The first line of input consists of a string that represents the name of the employee.

The second line consists of a double value that represents the rate for an hour.

The last line consists of an integer that represents the total hours worked.

Output Format

The output displays the total salary of the employee, where salary is rounded to two decimal places in the format: "Weekly Salary: Rs.<double value>".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: Dave

10.0

40

Output: Weekly Salary: Rs.400.00

Answer

```
import java.util.Scanner;
import java.text.DecimalFormat;

// You are using Java
class HourlyEmployee{
    private String name;
    private double rate;
    private int work;

    public HourlyEmployee(String name,double rate,int work){
        this.name = name;
        this.rate = rate;
        this.work = work;
    }
    public double calculateWeeklySalary(){
        if(work<=40){
            return work*rate;
        }
        else{
            double temp = 40*rate;
            work-=40;
            rate*=1.5;
            temp+=work*rate;
            return temp;
        }
    }
}
```

```

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

        String name = scanner.nextLine();
        double hourlyRate = scanner.nextDouble();
        int hoursWorked = scanner.nextInt();

        HourlyEmployee employee = new HourlyEmployee(name, hourlyRate,
hoursWorked);

        double weeklySalary = employee.calculateWeeklySalary();
        DecimalFormat df = new DecimalFormat("#.00");
        String formattedSalary = df.format(weeklySalary);
        System.out.println("Weekly Salary: Rs." + formattedSalary);
        scanner.close();
    }
}

```

Status : Correct

Marks : 10/10

4. Problem Statement

Mary is managing a business and wants to analyze its profitability. She operates both a regular business model and a seasonal business model. To assess profitability, she uses a program that calculates and compares the profit margins for both models based on revenue and cost.

The program defines:

BusinessUtility class with a method calculateMargin(double revenue, double cost). SeasonalBusinessUtility (inherits from BusinessUtility) and overrides calculateMargin(double revenue, double cost), adding a seasonal adjustment of 10% to the base margin. ProfitabilityChecker class with a method checkProfitability(double regularMargin), which prints "Business is profitable." if the regular margin is 10% or more, otherwise prints "Business is not profitable."

Mary inputs revenue and cost, and the program compute and display the regular and seasonal margins using:

$\text{Margin} = ((\text{Revenue} - \text{Cost}) / \text{Revenue}) \times 100$

$\text{Seasonal Margin} = \text{Margin} + 10$

Input Format

The first line of input consists of a double value *r*, representing the revenue.

The second line consists of a double value *c*, representing the cost.

Output Format

The first line prints a double value, representing the regular profit margin, rounded to two decimal places, in the format: "Regular Margin: X. XX%", where X.XX denotes the calculated regular margin.

The second line prints a double value, representing the seasonal profit margin, rounded to two decimal places, in the format: "Seasonal Margin: X. XX%", where X.XX denotes the calculated seasonal margin.

The third line prints a string, indicating whether the business is profitable or not profitable, based on the regular margin.

If the regular margin is less than 10, print "Business is not profitable.". If it is 10 or greater, print "Business is profitable."

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 1000.0

800.0

Output: Regular Margin: 20.00%

Seasonal Margin: 30.00%

Business is profitable.

Answer

```
import java.util.Scanner;
```

```
// You are using Java  
class BusinessUtility{
```

```
    public double calculateMargin(double revenue, double cost){  
        return((revenue - cost)/revenue)*100;  
    }  
}
```

```
class SeasonalBusinessUtility extends BusinessUtility{  
    @Override  
    public double calculateMargin(double revenue , double cost){  
        double base = super.calculateMargin(revenue,cost);  
        return base+10;  
    }  
}
```

```
class ProfitabilityChecker{  
    public void checkProfitability(double rm){  
        if(rm>=10){  
            System.out.println("Business is profitable.");  
        }  
        else{  
            System.out.println("Business is not profitable.");  
        }  
    }  
}
```

```
class Main {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        double revenue = scanner.nextDouble();  
        double cost = scanner.nextDouble();  
        BusinessUtility business = new BusinessUtility();  
        SeasonalBusinessUtility seasonalBusiness = new  
SeasonalBusinessUtility();  
        double regularMargin = business.calculateMargin(revenue, cost);  
        double seasonalMargin = seasonalBusiness.calculateMargin(revenue,  
cost);
```

```
        System.out.printf("Regular Margin: %.2f%%\n", regularMargin);  
        System.out.printf("Seasonal Margin: %.2f%%\n", seasonalMargin);
```

```
        ProfitabilityChecker checker = new ProfitabilityChecker();  
        checker.checkProfitability(regularMargin);  
        scanner.close();
```

```
    }  
}
```

Status : Correct

Marks : 10/10