

1. Create a Java class named Calculator with two methods:

i) multiply method that takes two integers and returns their product.

ii) multiply method overload that takes three doubles and returns their product.

Write a simple program to demonstrate the use of method overloading by calling both versions of the multiply method and printing the results.

```
import java.util.Scanner;

class Calculator {
    private int n1, n2;
    private double d1, d2, d3;

    public Calculator(int a, int b) {
        n1 = a;
        n2 = b;
    }

    public Calculator(double a, double b, double c) {
        d1 = a;
        d2 = b;
        d3 = c;
    }

    public int multiply(int a, int b) {
        return a * b;
    }

    public double multiply(double a, double b, double c) {
        return a * b * c;
    }
}

public class Main {
    public static void main(String[] args) {
        int choice, ans;
        Scanner scan = new Scanner(System.in);
        System.out.println("Enter your choice:");
        choice = scan.nextInt();

        switch (choice) {
            case 1: {
                int num1, num2;
                System.out.println("Enter num1 and num2 in int:");
                num1 = scan.nextInt();
                num2 = scan.nextInt();
                Calculator calc1 = new Calculator(num1, num2);
                ans = calc1.multiply(num1, num2);
                System.out.println("Result: " + ans);
                break;
            }
            case 2: {
                double num1, num2, num3;
                System.out.println("Enter num1, num2, and num3 in double:");
                num1 = scan.nextDouble();
```

```

        num2 = scan.nextDouble();
        num3 = scan.nextDouble();
        Calculator calc2 = new Calculator(num1, num2, num3);
        double result = calc2.multiply(num1, num2, num3);
        System.out.println("Result: " + result);
        break;
    }
    default:
        System.out.println("Invalid choice");
    }
}
}
}

```

Enter your choice:

1

Enter num1 and num2 in int:

3

90

Result: 270

Process finished with exit code 0

Enter your choice:

2

Enter num1, num2, and num3 in double:

2.5

6.2

4.12

Result: 63.86

Process finished with exit code 0

2 Create a class hierarchy representing different types of employees in a company. Design a base class Employee with fields for the employee's name, employee ID, and a method named calculateSalary() that returns the basic salary. Implement two subclasses: Manager and Developer.

Manager class should have an additional field for the bonus percentage. Developer class should have an additional field for the programming language.

Override the calculateSalary() method in both the Manager and Developer classes to include the bonus for managers and an extra allowance for developers. The basic salary for all employees is \$50,000.

Write a program to create instances of managers and developers, call the calculateSalary method on each, and print the details.

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

```
class Employee
```

```
{
```

```
    String empname;
```

```
    String empid;
```

```
    int basicpay;
```

```
    public Employee(String name,String id,int bpay)
```

```
    {
```

```
        empname = name;
```

```
        empid = id;
```

```
        basicpay = bpay;
```

```
    }
```

```
    public int calculateSalary() {
```

```
        return basicpay;
```

```
    }
```

```

}

class Manager extends Employee
{
    int bonus;
    public Manager(String name,String id,int bpay,int bon)
    {
        super(name,id,bpay);
        bonus = bon;
    }

    public int calculateSalary()
    {
        return (int) (basicpay + (basicpay * bonus/ 100));
    }

    public void displayDetails() {
        System.out.println("-----");
        System.out.println("Manager Details:");
        System.out.println("Name: " + empname);
        System.out.println("Employee ID: " + empid);
        System.out.println("Bonus Percentage: " + bonus + "%");
        System.out.println("Salary: $" + calculateSalary());
        System.out.println();
        System.out.println("-----");
    }
}

class Developer extends Employee
{
    int allowance;

    public Developer(String name,String id,int bpay,int allo)
    {
        super(name, id, bpay);
        allowance = allo;
    }

    public int calculateSalary()
    {
        return (int) (basicpay + allowance);
    }

    public void displayDetails() {
        System.out.println("-----");
        System.out.println("Developer Details:");
        System.out.println("Name: " + empname);
        System.out.println("Employee ID: " + empid);
        System.out.println("Allowance: $" + allowance);
        System.out.println("Salary: $" + calculateSalary());
        System.out.println();
        System.out.println("-----");
    }
}

public class Main {
    public static void main(String[] args) {
        String name;

```

```

String id;
Scanner scan = new Scanner(System.in);
/*System.out.println("Enter Employee name:");
name = scan.next();
System.out.println("Enter Employee id:");
id = scan.next();*/
int bpay = 50000;
//Employee emp = new Employee(name,id,bpay);
System.out.println("Enter the bonus percentage for Manager:");
int bon = scan.nextInt();
System.out.println("Enter allowance for developer:");
int allo = scan.nextInt();
Manager man = new Manager("Shobika","M001",bpay,bon);
Developer dev = new Developer("Priya","D001",bpay,allo);

man.displayDetails();
dev.displayDetails();
scan.close();
}
}

```

```

Enter the bonus percentage for Manager:

```

```

50

```

```

Enter allowance for developer:

```

```

2500

```

```

-----
Manager Details:

```

```

Name: Shobika

```

```

Employee ID: M001

```

```

Bonus Percentage: 50%

```

```

Salary: $75000

```

```

-----
Developer Details:

```

```

Name: Priya

```

```

Employee ID: D001

```

```

Allowance: $2500

```

```

Salary: $52500

```

```

-----
Process finished with exit code 0

```

3. Implement a class hierarchy with a base class Vehicle and two derived classes Car and Motorcycle.

The Vehicle class should have a method named calculateSpeed() that returns the speed of the vehicle. Override it in other two classes, where the speed is calculated as the product of the vehicle's speed and the number of passengers or wheels.

Note:

a) Car class should have an additional field for the number of passengers.

b) Motorcycle class should have an additional field for the number of wheels.

Write a program to create instances of car and motorcycle, call the calculateSpeed method on each, and determine the vehicle with the highest effective speed.

```
class Vehicle {
    int speed;

    public Vehicle(int speed) {
        this.speed = speed;
    }

    public int calculateSpeed() {
        return speed;
    }
}

class Car extends Vehicle {
    int numPassengers;

    public Car(int speed, int numPassengers) {
        super(speed);
        this.numPassengers = numPassengers;
    }

    @Override
    public int calculateSpeed() {

        return super.calculateSpeed() * numPassengers;
    }
}

class Motorcycle extends Vehicle {
    int numWheels;

    public Motorcycle(int speed, int numWheels) {
        super(speed);
        this.numWheels = numWheels;
    }

    @Override
    public int calculateSpeed() {

        return super.calculateSpeed() * numWheels;
    }
}
```

```

    }
}

public class Main {
    public static void main(String[] args) {

        Car car = new Car(60, 4); // Speed: 60, Passengers: 4
        Motorcycle motorcycle = new Motorcycle(80, 2); // Speed: 80, Wheels: 2

        int carSpeed = car.calculateSpeed();
        int motorcycleSpeed = motorcycle.calculateSpeed();

        String fastestVehicle;
        if (carSpeed > motorcycleSpeed) {
            fastestVehicle = "Car";
        } else if (carSpeed < motorcycleSpeed) {
            fastestVehicle = "Motorcycle";
        } else {
            fastestVehicle = "Both vehicles have the same effective speed.";
        }

        System.out.println("Car Speed: " + carSpeed);
        System.out.println("Motorcycle Speed: " + motorcycleSpeed);
        System.out.println("The vehicle with the highest effective speed is: " + fastestVehicle);
    }
}

```

```

Car Speed: 240
Motorcycle Speed: 160
The vehicle with the highest effective speed is: Car

Process finished with exit code 0

```