**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.**

**CODE :**

import java.util.Scanner;

public class Main {

public int multiply(int a, int b) {

return a \* b;

}

public double multiply(double x, double y, double z) {

return x \* y \* z;

}

public static void main(String[] args) {

Main calculator = new Main();

Scanner scanner = new Scanner(System.in);

System.out.print("Enter first integer: ");

int int1 = scanner.nextInt();

System.out.print("Enter second integer: ");

int int2 = scanner.nextInt();

int result1 = calculator.multiply(int1, int2);

System.out.println("Result of multiplying integers: " + result1);

System.out.print("Enter first double: ");

double double1 = scanner.nextDouble();

System.out.print("Enter second double: ");

double double2 = scanner.nextDouble();

System.out.print("Enter third double: ");

double double3 = scanner.nextDouble();

double result2 = calculator.multiply(double1, double2, double3);

System.out.println("Result of multiplying doubles: " + result2);

}

}

**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.**

**CODE :**

import java.util.Scanner;

class Employee {

private String name;

private int employeeID;

public Employee(String name, int employeeID) {

this.name = name;

this.employeeID = employeeID;

}

public double calculateSalary() {

return 50000.0;

}

public String getName() {

return name;

}

public int getEmployeeID() {

return employeeID;

}

}

class Manager extends Employee {

private double bonusPercentage;

public Manager(String name, int employeeID, double bonusPercentage) {

super(name, employeeID);

this.bonusPercentage = bonusPercentage;

}

@Override

public double calculateSalary() {

double basicSalary = super.calculateSalary();

return basicSalary + (basicSalary \* (bonusPercentage / 100));

}

public double getBonusPercentage() {

return bonusPercentage;

}

}

class Developer extends Employee {

private String programmingLanguage;

public Developer(String name, int employeeID, String programmingLanguage) {

super(name, employeeID);

this.programmingLanguage = programmingLanguage;

}

@Override

public double calculateSalary() {

double basicSalary = super.calculateSalary();

return basicSalary + 10000.0;

}

public String getProgrammingLanguage() {

return programmingLanguage;

}

}

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter Manager's name: ");

String managerName = scanner.nextLine();

System.out.print("Enter Manager's employee ID: ");

int managerID = scanner.nextInt();

System.out.print("Enter Manager's bonus percentage: ");

double bonusPercentage = scanner.nextDouble();

Manager manager = new Manager(managerName, managerID, bonusPercentage);

scanner.nextLine();

System.out.print("Enter Developer's name: ");

String developerName = scanner.nextLine();

System.out.print("Enter Developer's employee ID: ");

int developerID = scanner.nextInt();

Developer developer = new Developer(developerName, developerID, "Java");

System.out.println("Manager Details:");

System.out.println("Name: " + manager.getName());

System.out.println("Employee ID: " + manager.getEmployeeID());

System.out.println("Bonus Percentage: " + manager.getBonusPercentage() + "%");

System.out.println("Total Salary: $" + manager.calculateSalary());

System.out.println();

System.out.println("Developer Details:");

System.out.println("Name: " + developer.getName());

System.out.println("Employee ID: " + developer.getEmployeeID());

System.out.println("Programming Language: " + developer.getProgrammingLanguage());

System.out.println("Total Salary: $" + developer.calculateSalary());

}

}

**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.**

**CODE :**

import java.util.Scanner;

class Vehicle {

private double speed;

public Vehicle(double speed) {

this.speed = speed;

}

public double calculateSpeed() {

return speed;

}

}

class Car extends Vehicle {

private int numberOfPassengers;

public Car(double speed, int numberOfPassengers) {

super(speed);

this.numberOfPassengers = numberOfPassengers;

}

@Override

public double calculateSpeed() {

return super.calculateSpeed() \* numberOfPassengers;

}

public int getNumberOfPassengers() {

return numberOfPassengers;

}

}

class Motorcycle extends Vehicle {

private int numberOfWheels;

public Motorcycle(double speed, int numberOfWheels) {

super(speed);

this.numberOfWheels = numberOfWheels;

}

@Override

public double calculateSpeed() {

return super.calculateSpeed() \* numberOfWheels;

}

public int getNumberOfWheels() {

return numberOfWheels;

}

}

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter Car speed: ");

double carSpeed = scanner.nextDouble();

System.out.print("Enter the number of passengers in the Car: ");

int numberOfPassengers = scanner.nextInt();

Car car = new Car(carSpeed, numberOfPassengers);

System.out.print("Enter Motorcycle speed: ");

double motorcycleSpeed = scanner.nextDouble();

System.out.print("Enter the number of wheels on the Motorcycle: ");

int numberOfWheels = scanner.nextInt();

Motorcycle motorcycle = new Motorcycle(motorcycleSpeed, numberOfWheels);

double carEffectiveSpeed = car.calculateSpeed();

double motorcycleEffectiveSpeed = motorcycle.calculateSpeed();

System.out.println("Car Details:");

System.out.println("Speed: " + carSpeed);

System.out.println("Number of Passengers: " + numberOfPassengers);

System.out.println("Effective Speed: " + carEffectiveSpeed);

System.out.println();

System.out.println("Motorcycle Details:");

System.out.println("Speed: " + motorcycleSpeed);

System.out.println("Number of Wheels: " + numberOfWheels);

System.out.println("Effective Speed: " + motorcycleEffectiveSpeed);

System.out.println();

if (carEffectiveSpeed > motorcycleEffectiveSpeed) {

System.out.println("Car has the highest effective speed.");

} else if (carEffectiveSpeed < motorcycleEffectiveSpeed) {

System.out.println("Motorcycle has the highest effective speed.");

} else {

System.out.println("Both Car and Motorcycle have the same effective speed.");

}

}

}