Exp 5. Practicing Method Overloading and Constructor Overloading

**Overloading Questions**

1. **Question 1: Method Overloading in a Bank**  
   Write a Java class **BankAccount** with an overloaded method calculateInterest() that:
   * Calculates interest for a **savings account** (takes principal and rate as arguments).
   * Calculates interest for a **fixed deposit** (takes principal, rate, and time as arguments).
   * Calculates interest for a **current account** (no arguments, returns 0%).  
     Demonstrate all the overloaded methods in the main method.

class BankAccount{

double calculateInterest(int principal, double rate){

return ((double)principal\*rate)/100;

}

double calculateInterest(int principal, double rate, int time){

return ((double)principal\*rate\*time)/100;

}

double calculateInterest(){

return 0;

}

}

public class OverloadingBank {

public static void main(String[] args){

BankAccount b = new BankAccount();

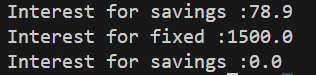
System.out.println("Interest for savings :"+ b.calculateInterest(1000, 7.89));

System.out.println("Interest for fixed :"+ b.calculateInterest(5000, 10, 3));

System.out.println("Interest for savings :"+ b.calculateInterest());

}

}



1. **Question 2: Overloading in a Geometry Context**  
   Create a class **Geometry** with overloaded methods area() that:
   * Computes the area of a circle (accepts radius).
   * Computes the area of a rectangle (accepts length and breadth).
   * Computes the area of a triangle (accepts base and height).  
     Test the methods in the main method.

class Geometery{

double area(int radius){

return 3.14\*radius\*radius;

}

int area(int length, int breadth){

return length\*breadth;

}

double area(double base, int height){

return (double)0.5\*base\*height;

}

}

public class GeometeryOverloading {

public static void main(String[] args) {

Geometery g = new Geometery();

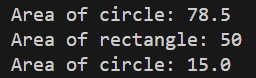
System.out.println("Area of circle: " + g.area(5));

System.out.println("Area of rectangle: " +g.area(5,10));

System.out.println("Area of circle: " +g.area(6.0, 5));

}

}



1. **Constructor Overloading**  
   Create a Java class named **Book** with the following constructor overloads:
2. A constructor that accepts **title** and **author**.
3. A constructor that accepts **title**, **author**, and **price**.
4. A constructor that accepts **title**, **author**, **price**, and **publisher**.  
   Use these constructors to create multiple objects and display their details.

class Book{

String title;

String author;

int price;

String publisher;

Book(String title, String author){

this.title = title;

this.author = author;

}

Book(String title, String author, int price){

this.title = title;

this.author = author;

this.price = price;

}

Book(String title, String author, int price, String publisher){

this.title = title;

this.author = author;

this.price = price;

this.publisher = publisher;

}

void display(){

System.out.println("Title: "+title+" Author: "+author+" Price: "+price+" Publisher: "+publisher);

}

}

public class BookOverloading {

public static void main(String[] args) {

Book b1 = new Book("H1", "Mani");

b1.display();

Book b2 = new Book("Joker", "Warner", 1000);

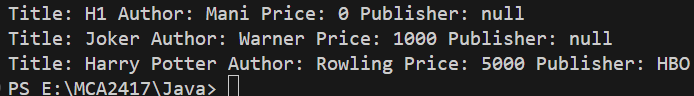
b2.display();

Book b3 = new Book("Harry Potter", "Rowling", 5000, "HBO");

b3.display();

}

}



**Packages**

**Creating and Using a Simple Package**

* Create a package named mathoperations containing a class Addition with a method add(int a, int b) that returns the sum of two numbers.
* Write another class TestAddition in a different package to import and use the Addition class.

**Using Multiple Classes in a Package**

* Create a package named geometry that contains two classes:
  + Circle: A class with a method area(double radius) to calculate the area of a circle.
  + Rectangle: A class with a method area(double length, double breadth) to calculate the area of a rectangle.
* Write a TestGeometry class in another package to use both classes.

**Package with Default and Parameterized Constructors**

* Create a package library containing a class Book with:
  + A default constructor that initializes a book's title as "Unknown".
  + A parameterized constructor to initialize a specific title.
* Add a method displayBook() to print the title.
* Test this package in another class by creating objects using both constructors.

**Student Management System**

* Create a package studentmanagement with two classes:
  + Student: Contains details like name, rollNumber, and grade.
  + StudentOperations: Contains methods to add, delete, and display students.
* Write a main program to test the functionality by importing the package.