**Assignment No.**: 3  
**Assignment Name**: Designing a Class with Constructors and Operator Overloading

**Java Program: Designing a Class with Constructors and Operator Overloading**

**Note**: Java does not support operator overloading directly like C++ does. However, we can simulate operator overloading by defining custom methods that perform operations on objects of the class. Below is an implementation of a class with constructors and an overloaded operator-like method.

// Class representing a complex number

class Complex {

private double real;

private double imaginary;

// Default constructor (sets default values)

public Complex() {

this.real = 0;

this.imaginary = 0;

}

// Parameterized constructor (sets values for real and imaginary parts)

public Complex(double real, double imaginary) {

this.real = real;

this.imaginary = imaginary;

}

// Method to simulate addition operator overloading

public Complex add(Complex other) {

double resultReal = this.real + other.real;

double resultImaginary = this.imaginary + other.imaginary;

return new Complex(resultReal, resultImaginary);

}

// Method to display the complex number

public void display() {

if (imaginary >= 0) {

System.out.println(real + " + " + imaginary + "i");

} else {

System.out.println(real + " - " + Math.abs(imaginary) + "i");

}

}

}

// Main class to test the Complex class and operator overloading

public class ComplexNumberDemo {

public static void main(String[] args) {

// Using default constructor

Complex c1 = new Complex();

System.out.print("Complex number 1 (default constructor): ");

c1.display();

// Using parameterized constructor

Complex c2 = new Complex(3.5, 2.5);

System.out.print("Complex number 2 (parameterized constructor): ");

c2.display();

// Adding complex numbers using the overloaded add method

Complex c3 = c1.add(c2);

System.out.print("Sum of c1 and c2: ");

c3.display();

// Another addition example

Complex c4 = new Complex(1.5, 4.5);

Complex c5 = new Complex(2.0, -1.0);

Complex c6 = c4.add(c5);

System.out.print("Sum of c4 and c5: ");

c6.display();

}

}

**Output:**

Complex number 1 (default constructor): 0.0 + 0.0i

Complex number 2 (parameterized constructor): 3.5 + 2.5i

Sum of c1 and c2: 3.5 + 2.5i

Sum of c4 and c5: 3.5 + 3.5i