Batch: A3	**Roll No.: 16010121045**
Experiment / assignment / tutorial No.01	
Grade: AA / AB / BB / BC / CC / CD /DD	
Signature of the Staff In-charge with date	

TITLE : Complex Arithmetic

AIM: Write a program to perform addition, subtraction, multiplication and division of two complex numbers. Demonstrate the use of classes and objects.
Variations :

Implementation of Program with One class

Accessibility with static and non-static methods within class and outside class.

Expected OUTCOME of Experiment:

CO2: Explore arrays, vectors, classes and objects in C++ and Java

Books/ Journals/ Websites referred:

1. E. Balagurusamy , "Programming with Java" McGraw-Hill.

2. Sachin Malhotra, Saurabh Choudhary, "Programming in Java", Oxford Publications.

Pre Lab/ Prior Concepts:

The Scanner class is a class in java.util, which allows the user to read values of various types. There are far more methods in class Scanner than you will need in this course. We only cover a small useful subset, ones that allow us to read in numeric values from either the keyboard or file without having to convert them from strings and determine if there are more values to be read.

Scanner in = new Scanner(System.in); // System.in is an InputStream
Numeric and String Methods

Method	Returns
int nextInt()	Returns the next token as an int. If the next token is not an integer,InputMismatchException is thrown.
long nextLong()	Returns the next token as a long. If the next token is not an integer,InputMismatchException is thrown.
float nextFloat()	Returns the next token as a float. If the next token is not a float or is out of range, InputMismatchException is thrown.
double nextDouble()	Returns the next token as a long. If the next token is not a float or is out of range, InputMismatchException is thrown.
String next()	Finds and returns the next complete token from this scanner and returns it as a string; a token is usually ended by whitespace such as a blank or line break. If not token exists,NoSuchElementException is thrown.
String nextLine()	Returns the rest of the current line, excluding any line separator at the end.
void close()	Closes the scanner.

The Scanner looks for tokens in the input. A token is a series of characters that ends with what Java calls whitespace. A whitespace character can be a blank, a tab character, a carriage return. Thus, if we read a line that has a series of numbers separated by blanks, the scanner will take each number as a separate token. .
The numeric values may all be on one line with blanks between each value or may be on separate lines. Whitespace characters (blanks or carriage returns) act as separators. The next method returns the next input value as a string, regardless of what is keyed. For example, given the following code segment and data

- int number = in.nextInt();
- float real = in.nextFloat();
- long number2 = in.nextLong();
- double real2 = in.nextDouble();
- String string = in.next();

Class Diagram:

Same Class:

Class name	exp1a
Variables	-
Functions	add (), sub(), multi(), div(), main()

Different Class:

Class name	exp1
Variables	-
Functions	main()

Class name	complex
Variables	r : double i : double
Functions	add (), sub(), multi(), div(), main()

Algorithm:

1. Start
2. Get input for num1
3. Get input for num2
4. Print addition : call add()
5. Print subtraction : call sub()
6. Print multiplication: call multi()
7. Print Division : call div()

Add()

1. Start
2. Add real numbers together
3. Add imaginary numbers together
4. Return the respective complex number

Sub()

1. Start
2. Subtract real numbers together
3. Subtract imaginary numbers together
4. Return the respective complex number

Multi()

1. Start
2. Multiply the given complex numbers using the mathematical formula for multiplication of complex numbers.
3. Return the respective complex number

Div()

1. Start
2. Divide the given complex numbers using the mathematical formula for Division of complex numbers.
3. Return the respective complex number

Implementation details:

Same Class:

```
import java.util.Scanner;

class exp1a {
    static void add(double r1,double i1,double r2,double i2)
{
        System.out.println("Addition Result:");
        System.out.println("The Real part is: " + (r1+r2) + "
Imaginary part is: " + (i1 + i2));
    }

    static void sub(double r1,double i1,double r2,double i2)
{
        System.out.println("Subtraction Result:");
        System.out.println("The Real part is: " + (r1 - r2) +
" Imaginary part is: " + (i1 - i2));
    }

    static void multi(double r1,double i1,double r2,double
i2) {
        System.out.println("Multiplication Result:");
        double real, img;
        real = (r1 * r2) - (i1 * i2);
        img = (r1 * i2) + (i1 * r2);
```

```
        System.out.println("The Real part is: " + real + "
Imaginary part is: " + img);
    }

    static void div(double r1,double i1,double r2,double i2)
{
        System.out.println("Division Result:");
        double den = (r2 * r2) + (i2 * i2);
        double real = (r1 * r2) + (i1 * i2);
        double img = (i1 * r2) - (r1 * i2);
        System.out.println("The Real part is: " + real / den
+ " Imaginary part is: " + img / den);
    }
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Real Part for num1: ");
        double real1 = sc.nextDouble();
        System.out.print("Enter Img Part for num1: ");
        double img1 = sc.nextDouble();
        System.out.print("Enter Real Part for num2: ");
        double real2 = sc.nextInt();
        System.out.print("Enter Img Part for num2: ");
        double img2 = sc.nextInt();
        exp1a.add(real1, img1, real2, img2);
        exp1a.sub(real1, img1, real2, img2);
        // Calling non-static methods
        exp1a.multi(real1, img1, real2, img2);
        exp1a.div(real1, img1, real2, img2);
        sc.close();
    }
}
```

Different Class:

```
import java.util.Scanner;

class exp1 {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Real Part for num1: ");
        double real = sc.nextDouble();
        System.out.print("Enter Img Part for num1: ");
        double img = sc.nextDouble();
        // Creating a complex number num1
        complex num1 = new complex(real, img);
        System.out.print("Enter Real Part for num2: ");
        real = sc.nextInt();
        System.out.print("Enter Img Part for num2: ");
        img = sc.nextInt();
        // Creating a complex number num2
        complex num2 = new complex(real, img);
        // Calling static methods
        complex.add(num1, num2);
        complex.sub(num1, num2);
        // Calling non-static methods
        num1.multi(num1, num2);
        num2.div(num1, num2);
        sc.close();
    }
}

class complex {
    double r, i;

    complex(double r, double i) {
        this.r = r;
        this.i = i;
    }

    static void add(complex num1, complex num2) {
        System.out.println("Addition Result:");
```

```
        System.out.println("The Real part is: " + (num1.r +
num2.r) + " Imaginary part is: " + (num1.i + num2.i));
    }

    static void sub(complex num1, complex num2) {
        System.out.println("Subtraction Result:");
        System.out.println("The Real part is: " + (num1.r -
num2.r) + " Imaginary part is: " + (num1.i - num2.i));
    }

    void multi(complex num1, complex num2) {
        System.out.println("Multiplication Result:");
        double real, img;
        real = (num1.r * num2.r) - (num1.i * num2.i);
        img = (num1.r * num2.i) + (num1.i * num2.r);
        System.out.println("The Real part is: " + real + "
Imaginary part is: " + img);
    }

    void div(complex num1, complex num2) {
        System.out.println("Division Result:");
        double den = (num2.r * num2.r) + (num2.i * num2.i);
        double real = (num1.r * num2.r) + (num1.i * num2.i);
        double img = (num1.i * num2.r) - (num1.r * num2.i);
        System.out.println("The Real part is: " + real / den
+ " Imaginary part is: " + img / den);
    }
}
```

Output:

Same Class:

```
pargat@Pargats-MacBook-Air OOPS % cd "/Users/pargat/Documents/COLLEGE/OOPS/"
Enter Real Part for num1: 10
Enter Img Part for num1: 5
Enter Real Part for num2: 20
Enter Img Part for num2: 4
Addition Result:
The Real part is: 30.0 Imaginary part is: 9.0
Subtraction Result:
The Real part is: -10.0 Imaginary part is: 1.0
Multiplication Result:
The Real part is: 180.0 Imaginary part is: 140.0
Division Result:
The Real part is: 0.5288461538461539 Imaginary part is: 0.14423076923076922
pargat@Pargats-MacBook-Air OOPS %
```

Different Class:

```
pargat@Pargats-MacBook-Air OOPS % cd "/Users/pargat/Documents/COLLEGE/OOPS/" &
Enter Real Part for num1: 10
Enter Img Part for num1: 2
Enter Real Part for num2: 20
Enter Img Part for num2: 1
Addition Result:
The Real part is: 30.0 Imaginary part is: 3.0
Subtraction Result:
The Real part is: -10.0 Imaginary part is: 1.0
Multiplication Result:
The Real part is: 198.0 Imaginary part is: 50.0
Division Result:
The Real part is: 0.5037406483790524 Imaginary part is: 0.07481296758104738
pargat@Pargats-MacBook-Air OOPS %
```

Conclusion:

In this experiment, we learnt how to implement program with one class and multiple classes. Along with accessibility with static and non-static methods within class and outside class.

Date: _____ **Signature of faculty in-charge**

Post Lab Descriptive Questions:

Q.1 Write a program to find the area and circumference of a circle using two classes.

Code:

```
import java.util.*;
class post1{
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.print(">>Enter Cirle Radius: ");
        double r=sc.nextDouble();
        circle.area(r);
        circle.circum(r);
    }
}
class circle{
    static void area(double r){
        System.out.println(">>The Area of circle is:
"+(3.14*r*r));
    }
    static void circum(double r){
        System.out.println(">>The Circumference of circle is:
"+(2*3.14*r));
    }
}
```

Output:

```
pargat@Pargats-MacBook-Air Practice % cd "/Users/pargat/Documents/C
>>Enter Cirle Radius: 15
>>The Area of circle is: 706.5
>>The Circumference of circle is: 94.2
pargat@Pargats-MacBook-Air Practice %
```

Q.2 Write the output of following program

```
1.    public class BreakExample2 {
2.    public static void main(String[] args) {
3.          //outer loop
4.          for(int i=1;i<=3;i++){
5.              //inner loop
6.              for(int j=1;j<=3;j++){
7.                 if(i==2&&j==2){
8.                    //using break statement inside the inner loop
9.                    break;
10.                }
11.                System.out.println(i+" "+j);
12.             }
13.         }
14.   }
15.   }
```

Output:

1 1

1 2

1 3

2 1

3 1

3 2

3 3

Q.3 Why is Java known as a platform independent language?

Java is based on Write-Once-Run-Anywhere concept that makes it Platform independent. It is platform independent because the program written in it is not directly converted into machine code but instead is converted into byte code by Java compiler, this byte code is then converted into machine readable code by Java Virtual Machine (JVM). JDK including JVM must be installed in a platform.

Q.4 Write a recursive static method for calculation of gcd of a number.

```
class post2 {
    public static void main(String[] args) {
        int a = 15, b = 150;
        System.out.println("G.C.D is: "+gcd(a,b));
    }
    public static int gcd(int a, int b) {
        if (b != 0)
            return gcd(b, a % b);
        else
            return a;
    }
}
```

Output:

```
pargat@Pargats-MacBook-Air Practice %
G.C.D is: 15
pargat@Pargats-MacBook-Air Practice %
```