LAB 4

Program to create a class complex with necessary operator overloading and type conversion.

Definitions

Private – An access specifier. Member data and member functions specified as private can't be accessed by methods outside the class. They can however be accessed by member functions.

String – String is a datatype. It is a sequence of characters. The ".split" method on a string is used to split a string into parts based on a given regular expression.

parseDouble() – This method in the Double class is used to convert a string into type double.

Constructor – It is a method in a class which is called during the creation of the class object. The constructor is usually used to initialize values of properties. It can also be used to call methods or output some value during object creation. A constructor does not need to be called separately; it is called during object creation by the compiler. The name of the constructor must be the same as the class name, and the constructor has no return type. The constructor can not be called by us, it is called automatically only when the object is created.

Overloading – Overloading is the use of one method to perform different actions. The methods have different signatures (return type, number of parameters, data type of parameters) but the same name.

Operator Overloading – Using an elementary operator, but overloading it to make it behave differently is operator overloading. This concept can't be used in java. Instead, we can define our own methods to act like the operator we want.

Code:

```
/* Program to create complex class
   with operator overloading and type conversion */

import java.util.Scanner;
import java.util.regex.Pattern;

class Complex {
   private double Re, Im;

   public Complex() {
      this(0, 0);
   }

   public Complex(String S) {
      String[] arr = S.split(Pattern.quote("+i"));
```

```
        this.Re = Double.parseDouble(arr[0]);
        this.Im = Double.parseDouble(arr[1]);
    }

    public Complex (int Re, int Im) {
        this((double)Re, (double)Im);
    }

    public Complex (double Re, double Im) {
        this.Re = Re;
        this.Im = Im;
    }

    public void printComplex() {
        System.out.printf("%.2f + %.2fi\n", this.Re, this.Im);
    }

    public static void sum(Complex A, Complex B) {
        Complex C = new Complex();
        C.Re = A.Re + B.Re;
        C.Im = A.Im + B.Im;

        C.printComplex();
    }

    public static void difference(Complex A, Complex B) {
        Complex C = new Complex();
        C.Re = A.Re - B.Re;
        C.Im = A.Im - B.Im;

        C.printComplex();
    }

    public static void product(Complex A, double x) {

        Complex C = new Complex(A.Re * x, A.Im * x);
        C.printComplex();
    }

    public static void product(Complex A, Complex B) {
        Complex C = new Complex();
        C.Re = A.Re * B.Re - A.Im * B.Im;
        C.Im = A.Re * B.Im - B.Re * A.Im;

        C.printComplex();
    }

    public static void quotient(Complex A, double x) {

        Complex C = new Complex(A.Re / x, A.Im / x);
        C.printComplex();
    }
```

```
   public static void quotient(Complex A, Complex B) {
      Complex C = new Complex();
      C.Re = (A.Re * B.Re - A.Im * B.Im) / (B.Re * B.Re + B.Im * B.Im);
      C.Im = (A.Re * B.Im + A.Im * B.Re) / (B.Re * B.Re + B.Im * B.Im);

      C.printComplex();
   }
}

public class Lab4 {
   public static void main(String[] args) {

      try(Scanner sc = new Scanner(System.in)) {

         //Creating a complex number using integer parameters
         System.out.print("Integer values of Re and Im : ");
         int Re1 = sc.nextInt();
         int Im1 = sc.nextInt();

         Complex C1 = new Complex(Re1, Im1);

         //Creating a complex number using double parameters
         System.out.print("Double values of Re and Im : ");
         double Re2 = sc.nextDouble();
         double Im2 = sc.nextDouble();

         Complex C2 = new Complex(Re2, Im2);

         //Creating a complex number using a string
         System.out.print("Type a complex number : ");
         String complex = sc.next();
         Complex C3 = new Complex(complex);
         System.out.println();

         //Printing the 3 complex numbers
         System.out.print("C1 : ");
         C1.printComplex();
         System.out.print("C2 : ");
         C2.printComplex();
         System.out.print("C3 : ");
         C3.printComplex();
         System.out.println();

         //Performing basic operations
         System.out.println("Sum and difference of C1 and C2 :");
         Complex.sum(C1, C2);
         Complex.difference(C1, C2);
         System.out.println();

         System.out.println("Product of C3 and 2.5 :");
         Complex.product(C3, 2.5);
```

```
        System.out.println("Product of C1 and C3 :");
        Complex.product(C1, C3);
        System.out.println();

        System.out.println("Quotient of C2 divided by 2.5 :");
        Complex.quotient(C2, 2.5);
        System.out.println("Quotient of C3 divided by C2 :");
        Complex.quotient(C3, C2);
        System.out.println();
    }
  }
}
```

Explanation:

- In this program, we have a class called constructor. This class creates a constructor and contains methods to perform elementary operations '+', '-', 'x', '/' on the complex numbers.
- We can initialize the constructor by entering integer values or double values for the real and imaginary parts, or by typing the constructor as a string.
- We can then perform operations on 2 constructors, or on a constructor and an integer/double.
- The methods for '+', '-', 'x', '/' have been declared as static. This means we don't need to initialize and object to call them. They can be called as

  ClassName.StaticMethod();

- We also have a printConstructor method to print the value of a constructor as a string.

Output:

```
PS D:\NITJ\Sem 4\Java\Lab 4> javac Lab4.java
PS D:\NITJ\Sem 4\Java\Lab 4> java Lab4
Integer values of Re and Im : 4 5
Double values of Re and Im : 3.2 1.4
Type a complex number : 2.4+i6.1

C1 : 4.00 + 5.00i
C2 : 3.20 + 1.40i
C3 : 2.40 + 6.10i

Sum and difference of C1 and C2 :
7.20 + 6.40i
0.80 + 3.60i

Product of C3 and 2.5 :
6.00 + 15.25i
Product of C1 and C3 :
-20.90 + 12.40i

Quotient of C2 divided by 2.5 :
1.28 + 0.56i
Quotient of C3 divided by C2 :
-0.07 + 1.88i

PS D:\NITJ\Sem 4\Java\Lab 4>
```