

|  |
| --- |
| LAB Assessment 01 |
|  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Presented to: | Muhammad Aksam Iftikhar | | **Course:** | Object Oriented Programming | | **Presented by:** | Asad Ali | | **Class:** | SP21-BCS-A | | **Registration ID:** | SP21-BCS-007 | | COMSATS  University Islamabad Lahore Campus |



**Complex** **Class**

import static java.lang.Math.\*;  
  
public class Complex {  
 private double real;  
 private double imaginary;  
  
 public Complex() {  
 setReal(0);  
 setImaginary(0);  
 }  
  
 public Complex(int real, int imaginary) {  
 setReal(real);  
 setImaginary(imaginary);  
 }  
  
 public Complex(Complex c) {  
 this.setReal(c.getReal());  
 this.setImaginary(c.getImaginary());  
 }  
  
 public double getReal() {  
 return real;  
 }  
  
 public void setReal(double real) {  
 this.real = real;  
 }  
  
 public double getImaginary() {  
 return imaginary;  
 }  
  
 public void setImaginary(double imaginary) {  
 this.imaginary = imaginary;  
 }  
  
 @Override  
 public String toString() {  
 return "( " + getReal() + " + " + getImaginary() + "i" + " )";  
 }  
  
 public double modulus() {  
 return sqrt((pow(getReal(), 2) + pow(getImaginary(), 2)));  
 }  
  
 public void addition(Complex c) {  
 this.setReal(this.getReal() + c.getReal());  
 this.setImaginary(this.getImaginary() + c.getImaginary());  
 }  
  
 public static Complex addition(Complex c1, Complex c2) {  
 Complex resultant = new Complex();  
  
 resultant.setReal(c1.getReal() + c2.getReal());  
 resultant.setImaginary(c1.getImaginary() + c2.getImaginary());  
  
 return resultant;  
 }  
  
 public void subtraction(Complex c) {  
 this.setReal(this.getReal() - c.getReal());  
 this.setImaginary(this.getImaginary() - c.getImaginary());  
 }  
  
 public static Complex subtraction(Complex c1, Complex c2) {  
 Complex resultant = new Complex();  
  
 resultant.setReal(c1.getReal() - c2.getReal());  
 resultant.setImaginary(c1.getImaginary() - c2.getImaginary());  
  
 return resultant;  
 }  
  
 public void multiplication(Complex c) {  
 double real = (this.getReal() \* c.getReal()) - (this.getImaginary() \* c.getImaginary());  
 double imaginary = (this.getReal() \* c.getImaginary()) + (this.getImaginary() \* c.getReal());  
  
 this.setReal(real);  
 this.setImaginary(imaginary);  
 }  
  
 public static Complex multiplication(Complex c1, Complex c2) {  
 Complex resultant = new Complex();  
  
 resultant.setReal((c1.getReal() \* c2.getReal()) - (c1.getImaginary() \* c2.getImaginary()));  
 resultant.setImaginary((c1.getReal() \* c2.getImaginary()) + (c1.getImaginary() \* c2.getReal()));  
  
 return resultant;  
 }  
  
 public void division(Complex c) {  
 Complex A = new Complex(this);  
  
 Complex T = constantMultiplication(A, (c.getReal() - c.getImaginary()));  
 double N = pow(c.getReal(), 2) - pow(c.getImaginary(), 2);  
  
 this.setReal(T.getReal() / N);  
 this.setImaginary(T.getImaginary() / N);  
 }  
  
 public static Complex division(Complex c1, Complex c2) {  
 Complex resultant = new Complex();  
  
 Complex T = constantMultiplication(c1, (c2.getReal() - c2.getImaginary()));  
 double N = (pow(c2.getReal(), 2)) - (pow(c2.getImaginary(), 2));  
  
 resultant.setReal(T.getReal() / N);  
 resultant.setImaginary(T.getImaginary() / N);  
  
 return resultant;  
 }  
  
 public static Complex constantMultiplication(Complex c, double constant) {  
 Complex resultant = new Complex();  
  
 resultant.setReal(c.getReal() \* constant);  
 resultant.setImaginary(c.getImaginary() \* constant);  
  
 return resultant;  
 }  
}

**ComplexTest** **Class**

import java.util.Scanner;  
  
public class ComplexTest {  
 public static void main(String[] args) {  
 Complex[] A = new Complex[3];  
 Complex[] B = new Complex[3];  
  
 A[0] = new Complex(5, 7);  
 A[1] = new Complex(3, 9);  
 A[2] = new Complex(2, -4);  
  
 B[0] = new Complex(1, 6);  
 B[1] = new Complex(7, -8);  
 B[2] = new Complex(5, 5);  
  
  
 for (int i = 0; i < A.length; i++) {  
 Complex resultant = Complex.addition(A[i],B[i]);  
 System.out.println("resultant = " + resultant);  
 System.out.println("Modulus of resultant = " + resultant.modulus());  
 System.out.println();  
 }  
 }  
}