

# Hw 7: The Complex Class

CS173- Intermediate Computing

**Due: April 22, 2019 at 23pm**

## Problem Description:

A complex number has the form  $a+bi$ , where  $a$  and  $b$  are real numbers and  $i$  is  $\sqrt{-1}$ . The numbers  $a$  and  $b$  are known as the real part and imaginary part of the complex number, respectively. You can perform addition, subtraction, multiplication, and division for complex numbers using the following formulas:

$$a+bi+c+di=(a+c)+(b+d)i$$

$$a+bi-(c+di)=(a-c)+(b-d)i$$

$$(a+bi)*(c+di)=(ac-bd)+(bc+ad)i$$

$$(a+bi)/(c+di)=(ac+bd)/(c^2+d^2)+(bc-ad)i/(c^2+d^2)$$

You can also obtain the absolute value for a complex number using the following formula:

$$|a+bi|=\sqrt{a^2+b^2}$$

Design a class named Complex for representing complex numbers and the functions add, subtract, multiply, divide, abs for performing complex-number operations, and the toString function for returning a string representation for a complex number. The toString function returns a + bi as a string. If b is 0, it simply returns a.

Provide three constructors Complex(a, b), Complex(a), and Complex(). Complex() creates a Complex object for number 0 and Complex(a) creates a Complex object with 0 for b. Also provide the getRealPart() and getImaginaryPart() methods for returning the real and imaginary part of the complex number, respectively.

Overload the operators +, -, \*, /, +=, -=, \*=, /=, [], unary + and -, prefix ++ and --, postfix ++ and --, <<, >>.

Overload the operators +, -, \*, /, <<, and >> as nonmember functions. Overload [] so that [0] returns a and [1] returns b.

Use the following main function to test your program.

```

int main()
{
    Complex number1;
    cout << "Enter the first complex number: ";
    cin >> number1;

    Complex number2;
    cout << "Enter the second complex number: ";
    cin >> number2;

    cout << "(" << number1 << ")" << " + " << "(" << number2
        << ")" << " = " << (number1 + number2) << endl;
    cout << "(" << number1 << ")" << " - " << "(" << number2
        << ")" << " = " << (number1 - number2) << endl;
    cout << "(" << number1 << ")" << " * " << "(" << number2
        << ")" << " = " << (number1 * number2) << endl;
    cout << "(" << number1 << ")" << " / " << "(" << number2
        << ")" << " = " << (number1 / number2) << endl;
    cout << "| " << number1 << " | " << " = " << number1.abs() << endl;

    number1[0] = 3.4;
    cout << number1++ << endl;
    cout << ++number2 << endl;
    cout << (3 + number2) << endl;
    cout << (number2 += number1) << endl;
    cout << (number2 *= number1) << endl;

    return 0;
}

```

Design:  
 (Draw the UML diagram for the Complex class)