CPP Problem Design

Subject: Complex

Contributor: 林承達, 陳俊儒, 廖宣瑋

Main testing concept: Operator overloading

Basics

■ C++ BASICS
□ FLOW OF CONTROL
■ FUNCTION BASICS
□ PARAMETERS AND OVERLOADING
□ ARRAYS
■ STRUCTURES AND CLASSES
■ CONSTRUCTORS AND OTHER TOOLS
■ OPERATOR OVERLOADING, FRIENDS, AND REFERENCES
□ STRINGS
□ POINTERS AND DYNAMIC ARRAYS

Functions

□ SEPARATE COMPILATION AND NAMESPACES
□ STREAMS AND FILE I/O
□ RECURSION
□ INHERITANCE
□ POLYMORPHISM AND VIRTUAL FUNCTIONS
□ TEMPLATES
□ LINKED DATA STRUCTURES
□ EXCEPTION HANDLING
□ STANDARD TEMPLATE LIBRARY
□ PATTERNS AND UML

Description:

Define a class for complex numbers. A complex number is a number of the form as following:
a+b*i
b+c*i
where for our purposes, a and b are numbers of type double, and i is a number that represents the quantity $\sqrt{-1}$. Represent a complex number as two values of type double. Name the member variable **realValue** and **imaginaryValue**. (The variable for the number that is multiplied by i is the one called imaginary)
Define a class named **Complex**. Include a constructor with two parameters of type double that can be used to set the member variables of an object to any values. Include another constructor that has only a single parameter of type double; call this parameter **readPart** and define the constructor so that the object will be initialized to realPart + 0*i. Include a default constructor that initializes an object to 0(0 + 0i).
Using real() to get **realValue** and imag() to get **imaginaryValue**. By the way, you have to define norm() to get the norm of complex which defined as $\sqrt{a^2+b^2}$.
Overload all the following operator so that they can apply to the Complex : ==, +, -, *, >>, and << correctly. You should write a test program to test your class.

Input:

No input.

Output:

As the following sample.

Sample Input / Output：

Sample Input	Sample Output

#include"complex.h"
int main()
{
        // test constructors
        Complex x, y(3), z(-3.2, 2.1);
	x = 0 + 0*i
	y = 3 + 0*i
	z = -3.2 + 2.1*i

```
        cout << "x =  " << x << "y = " << y
<< "z = " << z << endl << endl;

        x = Complex(3, -4);

        cout << "testing members and support
functions as well as"
                << " output operator:\n"
                << "complex number x = " << x
<< endl
                << "real part: " << x.real()
<< endl
                << "real part from friend
real(x): " << real(x) << endl
                << "imaginary part: " <<
x.imag() << endl
                << "imaginary part from friend
imag(x) : " << imag(x) << endl
                << "norm: " << norm(x) << endl
<< endl;

        cout << "test operator ==:" << endl
<< endl;
        if (x == y)
                cout << "x = y" << endl <<
endl;
        else
                cout << "x!=y" << endl <<
endl;

        cout << "test complex arithmetic and
output routines: \n\n";
        y = Complex(1, -1);
        cout << "x =  " << x << "y = " << y
<< "z = " << z << endl << endl;

        z = x + y;
        cout << "z = x + y = " << z << endl;

        z = x * y;
        cout << "z = x * y = " << z << endl;

        z = x - y;
        cout << "z = x - y = " << z << endl;

        z = x / y;
        cout << "z = x / y = " << z << endl
<< endl;

        //test of automatic conversion double
-> complex by the constructor.

        double d(2.0);
        cout << "d: " << d << "   x: " << x
<< endl;
        cout << "x+d: ";
        z = x + d;
        cout << z << endl;
        z = x - d;
        cout << "x-d: ";
        cout << z << endl;
        z = x * d;
        cout << "x*d: ";
        cout << z << endl;
        z = x / d;
```

```
testing members and
support functions as well
as output operator:
complex number x = 3 + -
4*i

real part: 3
real part from friend
real(x): 3
imaginary part: -4
imaginary part from friend
imag(x) : -4
norm: 5

test operator ==:

x!=y

test complex arithmetic
and output routines:

x =  3 + -4*i
y = 1 + -1*i
z = -3.2 + 2.1*i


z = x + y = 4 + -5*i

z = x * y = -1 + -7*i

z = x - y = 2 + -3*i

z = x / y = 3.5 + -0.5*i


d: 2   x: 3 + -4*i

x+d: 5 + -4*i

x-d: 1 + -4*i

x*d: 6 + -8*i

x/d: 1.5 + -2*i

d+x: 5 + -4*i

d-x: -1 + 4*i

d*x: 6 + -8*i

d/x: 0.24 + 0.32*i

two/x: 0.24 + 0.32*i


Getting data from standard
input:
data read is:
x = 3 + 4*i
y = 5 + 6*i
```

```
        cout << "x/d: ";
        cout << z << endl;
        z = d + x;
        cout << "d+x: ";
        cout << z << endl;
        z = d - x;
        cout << "d-x: ";
        cout << z << endl;
        z = d * x;
        cout << "d*x: ";;
        cout << z << endl;
        z = d / x;
        cout << "d/x: ";;
        cout << z << endl;

        //test whether double/complex and
complex/complex give same result:
        Complex two(2, 0);
        cout << "two/x: ";
        cout << two / x << endl;

        cout << "\nGetting data from standard
input: \n";
        cin >> x >> y;
        cout << "data read is: x = " << x <<
" y = " << y << endl << endl;
        return 0;
}
```

□ Eazy,Only basic programming syntax and structure are required.

■ Medium,Multiple programming grammars and structures are required.

□ Hard,Need to use multiple program structures or more complex data types.

Expected solving time:

30 minutes

Other notes: