

Object-Oriented Programming, Fall 2020

Homework Assignment #3

Due midnight Wednesday, November 11, 2020

Instructions

1. If any question is unclear, please ask for a clarification.
2. You may try to reuse as much of the source code supplemented as possible.
3. Unless stated otherwise, all the line numbers for the program listings are for reference only.
4. You are required to do all the homework assignments on Linux and use g++ version 10.2.0 or later.
5. You are required to give your TA a demo of your program. Make sure that your program can compile and run on the server machine, which will be used for the demo.
6. For the program that you write, you are required to include a Makefile. Otherwise, your homework will not be graded—meaning that you will receive zero marks.
7. Unless stated otherwise, you are required to work on the homework assignment individually.
8. No late homework will be accepted.

Programming Project

This assignment requires that you finish the implementation of the **Complex** class, the interface of which is as defined in Listing 1. To make it easier for you to test your implementation, also given are the driver program and the output of the driver program (see Listings 2 and 3, the source code of which are also available in /student/data/hw3). Your implementation of the **Complex** class has to be in a separate file named Complex.cpp.

Listing 1: Complex.h

```
1 #ifndef __complex_h_included__
2 #define __complex_h_included__
3
4 #include <iostream>
5
6 using std::ostream;
7
8 class Complex {
9 public:
```

```

10     Complex(const double re = 0, const double im = 0);
11     Complex(const Complex& c);
12     Complex& operator=(const Complex& c);
13
14     Complex Polar(const double leng, const double arg);
15
16     double Real();
17     double Imag();
18     double Norm();
19     double Abs();
20     double Arg();
21
22     Complex operator++();
23     Complex operator++(int);
24     Complex operator--();
25     Complex operator--(int);
26
27     ~Complex();
28
29     friend Complex Polar(const double leng, const double arg);
30     friend double Norm(const Complex& x);
31     friend double Abs(const Complex& x);
32     friend double Arg(const Complex& x);
33     friend Complex operator+(const Complex& x, const Complex& y);
34     friend Complex operator-(const Complex& x, const Complex& y);
35     friend Complex operator*(const Complex& x, const Complex& y);
36     friend Complex operator/(const Complex& x, const Complex& y);
37     friend Complex operator+=(Complex& x, const Complex& y);
38     friend Complex operator-=(Complex& x, const Complex& y);
39     friend Complex operator*=(Complex& x, const Complex& y);
40     friend Complex operator/=(Complex& x, const Complex& y);
41     friend bool operator==(const Complex& x, const Complex& y);
42     friend bool operator!=(const Complex& x, const Complex& y);
43     friend ostream& operator<<(ostream& o, const Complex& x);
44
45 private:
46     double real;
47     double imag;
48 };
49
50 #endif

```

Listing 2: main.cpp

```

1  #include <iostream>
2  #include "Complex.h"
3
4  using std::cout;
5  using std::endl;
6
7  int main()
8  {
9      Complex a(77, 66.3);
10     Complex b(a);
11     int i = 1;
12
13     cout << i++ << ": (" << a.Real() << ', ' << a.Imag() << ') ' << endl;
14     cout << i++ << ": " << b << endl;
15     cout << i++ << ": " << Norm(b) << endl;
16     cout << i++ << ": " << b.Norm() << endl;
17     cout << i++ << ": " << Abs(b) << endl;
18     cout << i++ << ": " << b.Abs() << endl;
19     cout << i++ << ": " << Arg(b) << endl;
20     cout << i++ << ": " << b.Arg() << endl;

```

```

21
22     a = Complex(12, 33.2);
23     cout << i++ << ": " << a << endl;
24     cout << i++ << ": " << 1+a << endl;
25
26     cout << i++ << ": " << a++ << endl;
27     cout << i++ << ": " << ++a << endl;
28
29     b = a.Polar(5.6, 1.8);
30
31     cout << i++ << ": " << a << endl;
32     cout << i++ << ": " << b << endl;
33
34     b = Polar(6.5, 8.1);
35     cout << i++ << ": " << b << endl;
36
37     cout << i++ << ": " << a+b << endl;
38     cout << i++ << ": " << a-b << endl;
39     cout << i++ << ": " << a*b << endl;
40     cout << i++ << ": " << a/b << endl;
41
42     a /= b;
43     cout << i++ << ": " << a << endl;
44     a /= Complex(4, 3);
45     cout << i++ << ": " << a << endl;
46
47     a *= b;
48     cout << i++ << ": " << a << endl;
49     a *= Polar(5.1, 5.1);
50     cout << i++ << ": " << a << endl;
51
52     a += b;
53     b -= a;
54     cout << i++ << ": " << b << endl;
55
56     if (!(a == b))
57         cout << i++ << ": " << "OK" << endl;
58     if (a != b)
59         cout << i++ << ": " << "OK" << endl;
60 }

```

Listing 3: complex.out

```

1 1: (77,66.3)
2 2: (77,66.3)
3 3: 10324.7
4 4: 10324.7
5 5: 101.61
6 6: 101.61
7 7: 0.710868
8 8: 0.710868
9 9: (12,33.2)
10 10: (13,33.2)
11 11: (12,33.2)
12 12: (14,35.2)
13 13: (-1.27233,5.45355)
14 14: (-1.27233,5.45355)
15 15: (-1.58304,6.30428)
16 16: (-2.85537,11.7578)
17 17: (0.310705,-0.850737)
18 18: (-32.3666,-16.6543)
19 19: (0.861417,-0.0144858)
20 20: (0.861417,-0.0144858)
21 21: (0.136088,-0.105688)

```

```
22 22: (0.450853,1.02525)
23 23: (5.70997,-0.152415)
24 24: (-5.70997,0.152415)
25 25: OK
26 26: OK
```

Grading Policy

The grading policy for this assignment is as follows:

- Make sure that a **Makefile**, which contains at least three targets—**all**, **dep**, and **clean**—is provided. Otherwise, the grade for your program will be zero.
- 80 points if your program compiles without errors and warnings, and the answer is correct.
- 20 points if you use only C++ Streams and File I/O—as given in Chapter 12 of the text—for all the I/Os. In other words, *no* C library function calls such as **open**, **close**, **read**, **write**, the **scanf** family of functions, the **printf** family of functions, are used in your program.