

Computing 2 - Labs

Lab 6: Complex Class Overloaded

Consider `class Complex` shown below. The class enables operations on complex numbers.

1. Overload the addition, subtraction, and multiplication operator to enable these arithmetic operations of two complex numbers as in algebra.
2. Modify the class to enable input and output of complex numbers via overloaded `>>` and `<<` operators, respectively.
3. Overload the `==` and `!=` operators to allow comparisons of complex numbers. Comparisons return Boolean numbers. The underlying assumptions are either met or not, and the function then returns one or zero, respectively.

The user input is to be provided as "(real, imaginary)". Although there could be an arbitrary amount of white space, e.g., resulting in an input of "(real, imaginary)", there is no need to consider this here, but you could. Turn to the lecture for details on how to skip white space, e.g., using `ws`. The public member function `std::istream::ignore` may be useful to ignore parentheses and the comma. By default, it extracts one character (the first one) from the input sequence and discards it. Using `ignore(2)` means 2 characters in the input sequence shall be ignored.

When overloading the `<<` operator, format the output such that complex numbers are displayed as "(real, imaginary)".

Here is the class definition:

```
class Complex
{
    friend std::ostream &operator<<(std::ostream &, const Complex &);
    friend std::istream &operator>>(std::istream &, Complex &);
public:
    explicit Complex( double = 0.0, double = 0.0 ); // constructor
    Complex operator+( const Complex& ) const; // addition
    Complex operator-( const Complex& ) const; // subtraction
    Complex operator*(const Complex&) const; // multiplication
    bool operator==(const Complex&) const; // bool equality
    bool operator!=(const Complex&) const; // bool inequality
private:
    double real_; // real part
    double imag_; // imaginary part
}; // end class Complex
```

Here is the driver program:

```
int main()
{
    Complex x, y(4.3, 8.2), z(3.3, 1.1), k;

    cout << "Enter a complex number as (real, imaginary) followed by ENTER? ";
    cin >> k; // demonstrating overloaded >>
    cout << "x: " << x << "\ny: " << y << "\nz: " << z << "\nk: "
        << k << '\n'; // demonstrating overloaded <<
    x = y + z; // demonstrating overloaded +
    cout << "\nx = y + z:\n" << x << " = " << y << " + " << z << '\n';
    x = y - z; // demonstrating overloaded -
    cout << "\nx = y - z:\n" << x << " = " << y << " - " << z << '\n';
    x = y * z; // demonstrating overloaded *
    cout << "\nx = y * z:\n" << x << " = " << y << " * " << z << "\n\n";

    if (x != k) // demonstrating overloaded !=
        cout << x << " != " << k << '\n';

    cout << '\n';
    x = k;

    if (x == k) // demonstrating overloaded ==
        cout << x << " == " << k << '\n';
} // end main
```

Given (1, 2) as input, it produces the following output:

```
Enter a complex number as (real, imaginary) followed by ENTER? <1,2>
x: <0, 0>
y: <4.3, 8.2>
z: <3.3, 1.1>
k: <1, 2>

x = y + z:
<7.6, 9.3> = <4.3, 8.2> + <3.3, 1.1>

x = y - z:
<1, 7.1> = <4.3, 8.2> - <3.3, 1.1>

x = y * z:
<5.17, 31.79> = <4.3, 8.2> * <3.3, 1.1>

<5.17, 31.79> != <1, 2>

<1, 2> == <1, 2>
```

Task:

Given the header file and the driver file, implement the definitions of the class member functions.