Programming Assignment 5: Complex Number Overloading

The **complex number** a+bi, contains two real numbers, the *real part* a and the *imaginary part* b. Write a C++ project that defines **class** Complex of complex numbers with overloading operations. The class contains two constructors:

1. Default constructor with default value 0.0+0.0i.
2. Copy constructor of a Complex parameter.

There are five complex number arithmetic operations:

1. Complex addition: (a+bi)+(c+di) = (a+c)+(b d)i
2. Complex subtraction: (a+bi)-(c+di) = (a-c)+(b-d)i
3. Complex multiplication: (a+bi)×(c+di) = (a×c-b×d) + (a×d+b×c)i
4. Complex division: (a+bi)÷(c+di) = ((a×c+b×d)+(-a×d+b×c)i)÷(c^2+d^2)
5. Complex absolute value: |a+bi| = $(a^2+b^2)^{1/2}$

Also, there are four supporting methods:

1. Get real part of a complex number: getRe()
2. Get imaginary part of a complex number: getIm()
3. Set real part of a complex number: setRe(r)
4. Set imaginary part of a complex number: setIm(s)

Use overloaded operator member functions and friend functions to define and implement arithmetic operations a+b (addition), a-b (subtraction), a×b (multiplication), a÷b (division), and and comparing relations a==b (equal to), a!=b (not equal to), where a and b is either a real number or a complex number, but not both of them are real numbers.

Use overloaded operator member functions to define and implement assignment statement of complex numbers and its variations, including, ⟨variable⟩ = ⟨expression⟩, ⟨variable⟩ += ⟨expression⟩, ⟨variable⟩ −= ⟨expression⟩, ⟨variable⟩ ∗= ⟨expression⟩, and ⟨variable⟩ /= ⟨expression⟩. The left-hand-side ⟨variable⟩ is a complex reference and the right-hand-side ⟨expression⟩ is either a **double** object or a complex object.

Overload I/O stream operators << and >> to enable output and input of complex numbers.

Use **class** Complex to write an application program to solve a quadratic equation and verify the two roots. In your output, print a real numeral four digits after the decimal point. When verify the two roots, consider the precision error up to six digits after the decimal point, *i.e.*, the absolute value the result of substituting a root to the quadratic equation is less than 0.000001.

The Dev++ project files are **complex_overloading.dev**, **complex_overloading.h**, **complex_overloading.cpp**, and **quadratic_equation_verifier.cpp**. Compress the **.dev**, **.h**, and **.cpp** program files in a file named **Assignment_5_DXXXXXXX.YYY**, where **YYY** is **zip**, **rar**, or **7z**. Write an assignment report **Report_5_DXXXXXXX.pdf**. In the report, (1) explain the differences in programming complex number assignments using C and C++ and (2) discuss the advantages and/or disadvantages of programming in C++. Submit the assignment report **Report_5_DXXXXXXX.pdf** and the compressed program file **Assignment_5_DXXXXXXX.yyy**, where **DXXXXXXX** is your student ID. The assignment is due by **23:59 pm, Wednesday, May 15, 2024**.

Add #include <iomanip> and "cout << fixed << setprecision(4);" to print **double** type data in the fixed point format and 4 digits after the decimal point. Program execution example:

命令提示符

```
D:\>complex_overloading
Enter coefficients a, b, and c: 2.0 4.0 -6.0

The two roots of quadratic equation 2.0000X**2+4.0000X-6.0000=0.0000 are:
        1.0000 and -3.0000
Verification of the two quadratic equation roots PASSES.

D:\>complex_overloading
Enter coefficients a, b, and c: -3 5 -4

The two roots of quadratic equation -3.0000X**2+5.0000X-4.0000=0.0000 are:
        0.8333-0.7993i and 0.8333+0.7993i
Verification of the two quadratic equation roots PASSES.

D:\>complex_overloading
Enter coefficients a, b, and c: 1.0 -1.0 3.2

The two roots of quadratic equation X**2-X+3.2000=0.0000 are:
        0.5000+1.7176i and 0.5000-1.7176i
Verification of the two quadratic equation roots PASSES.

D:\>
```