1 Operator Overloading, Programming Challenges III

Create a class called **Complex** for performing arithmetic with complex numbers. Complex numbers have the form

$$realpart + imaginarypart * i$$

where **i** is

$$\sqrt{-1}$$

Use floating–point variables to represent the **private** data of the class. Provide a constructor function that enables an object of this class to be initialized when it is declared. The constructor should contain default values 0.0 in case no initializers are provided. You should do the following things:

- Create a copy constructor for class **Complex**.
- Overload the >> operator to input a complex number like a+bi in the form (a, b) and overload the << operator to output the same complex number in the form (a, b)
- Overload the == operator to compare two complex number for equality. This function should return a **bool**ean value.
- Overload the += operator for the **Complex** class.
- Overload the postincrement and preincrement operators for the class Complex. When a complex number is incremented, only real part of the complex number should be increased by one.
- Overload the postdecrement and predecrement operators for the class Complex. When a complex number is decreased, only real part of the complex number should be decreased by one.
- Overload the addition operator to enable addition of two complex numbers as in algebra.
- Overload the subtraction operator to enable subtraction of two complex numbers as in algebra.
- Overload the multiplication operator to enable addition of two complex numbers as in algebra. Complex number multiplication is performed as follows:

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

- Overload the division operator to enable addition of two complex numbers as in algebra. Complex number division is performed as follows:

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

Tips:

- When overloading the stream-extraction (>>) operator, use the **ignore** method of the class **istream**.
- Overloaded << and >> should be **friend** functions.
- All overloaded operators should be **const** functions.