

# Lab 6: Operator Overloading in C++

## Topics

1. Classes
2. Operator overloading
3. Friend functions
4. Member initializer lists

## Introduction

In this lab, you will be implementing a complex number class. This class is to be implemented such that it supports the use of familiar “native” operators such as ‘+’, ‘-’, ‘\*’, ‘/’ etc.

## Tasks

Setup your complex class as follows

```
class Cmplx {  
    private :  
        double real, imag; // real and imaginary part of cmplx number  
    public :  
        Cmplx () : real (0.0), imag (0.0) //Default Constructor  
        {}  
        Cmplx (const double r, const double i) : real (r), imag (i)  
        {}  
        ~Cmplx() // Destructor  
        {}  
        /*  
        * NB: additional function declarations go in here!  
        */  
};
```

1. Overload the '+' addition operator such that you can perform the addition of two complex numbers. That is, if 'a' and 'b' are objects of the complex class, the two can be added like this: `a + b`.
2. Overload the '-' subtraction operator such that you can perform the subtraction of two complex numbers. That is, if 'a' and 'b' are objects of the complex class, the difference of the two can be computed like this: `a - b`.
3. Overload the "<<" insertion operator such that you can utilize the C++ output stream to print out an object of your complex class. That is, if 'a' is an object of the complex class, it can be printed out like this: `std::cout << a << std::endl`.
4. Overload the ">>" extraction operator such that you can utilize the C++ input stream to read in an object of your complex class. That is, if 'a' is an object of the complex class, values can be stored in it like this: `std::cin >> a`.

## Notes

1. When you are printing out the complex number, you should print it out in the rectangular form, i.e. `a + bi`
2. When you are reading in the complex number, you should read it in the comma separated form, i.e. `(a, b)`.
3. C++ 'friend' functions:
  - Overloaded output stream output operator '<<' for use with the Cmplx class. `std::ostream & operator << ( const Cmplx &cmplx_num );`
  - Overloaded input stream input operator '>>' for use with the Cmplx class. `std::istream & operator >> ( Cmplx & cmplx_num );`
  - Overloaded addition operator '+' for use with the Cmplx class. `Cmplx operator + (const Cmplx &lhs, const Cmplx &rhs)`
  - Overloaded subtraction operator '-' for use with the Cmplx class. `Cmplx operator - (const Cmplx &lhs, const Cmplx &rhs)`

## Deliverables

Create a gzipped tarball file of your ~/ece3220-lab06-PAWPRINT directory and upload that tarball via the Submit Assignment link for this lab. This should include your source file(s) and Makefile. If you are unable to create a tarball file, a

simple zip file will suffice. You may also submit all your multiple files individually if needed.