

University of Nevada, Las Vegas Computer Science 302 Spring 2017

Assignment 1: Due September 7, 2018

C++ complex class

You are permitted to work in groups, get help from others, read books, and use the internet. You may not copy code electronically, nor share your code electronically with any other student.

- Write a class for complex numbers.
- Submit three files to the graduate assistant, Shradha Kapoor, by midnight Friday September 5.
- The three files are `cmplx.h`, the header file, `cmplx.cxx` (or `cmplx.cpp`) which implements the members of the class, and a main file, which you could call `main.cxx` or `main.cpp` or something else if you prefer.
- You are not required to implement any of the following member functions:

1. `modulus()`
2. `abssqr()`
3. `sqrt()`
4. `exp()`

Those functions are here for future expansion.

- Your program must execute as follows.
 1. Prompt the user to enter a complex number z .
 2. Read z from the input file. (You will need to decide how to enter a complex number.)
 3. Write z , the conjugate of z , $z - 3$, $2 + z$, $3/z$, z^2 .
 4. Prompt the user to enter a complex number w .
 5. Read w from the input file.
 6. Write w , the conjugate of w , $z + w$, $z - w$, $z * w$, z/w .
 7. for the first of the three iterations, let $z = -i$ and $w = 1 + i$. The the other two iterations, pick other values of z and w . Your choices should differ from the choices of any other student.
- Your variable names should not be z and w . In fact, they should be different from any other student's variable names.
- Your output should look nice. For example, $-i$ should be output as `-i` and not as `0+-1i`.
- Here is my version of the header file:

```

class cmplx
{
    friend ostream& operator <<(ostream&outs,const cmplx&) // Error was HERE
    friend istream& operator >>(istream&ins, cmplx&);
    friend cmplx operator+(double,const cmplx&);
    friend cmplx operator-(double,const cmplx&);
    friend cmplx operator*(double,const cmplx&);
    friend cmplx operator/(double,const cmplx&);
    friend bool operator==(double,const cmplx&);
    friend bool operator!=(double,const cmplx&);
    friend cmplx sqrt(const cmplx&);
    friend cmplx exp(const cmplx&);
public:
    static const cmplx i; // the imaginary square root of -1
    cmplx(double a,double b); // we write a complex number as a+bi
    cmplx();
    const cmplx & operator=(const cmplx&);
    const cmplx & operator=(double); // changes real number into complex
    double getr() const {return rpart;}
    double geti() const {return ipart;}
    cmplx conjugate() const; // the conjugate of a+bi is a-bi
    cmplx operator!() const; // !z means the conjugate of z
    cmplx operator+(const cmplx&) const;
    cmplx operator+(double) const;
    cmplx operator-() const; // additive inverse
    cmplx operator-(const cmplx&) const;
    cmplx operator-(double) const;
    cmplx operator*(const cmplx&) const;
    cmplx operator*(double) const;
    bool operator==(const cmplx&) const;
    bool operator==(double) const;
    bool operator!=(const cmplx&) const;
    bool operator!=(double) const;
    double abssqr() const; // the square of the absolute value
    double modulus() const; // the absolute value
    cmplx inverse() const; // multiplicative inverse, i.e. 1/z
    cmplx operator/(const cmplx&) const;
    cmplx operator/(double) const;
    //cmplx sqrt() const; // principle value of the square root
    //cmplx exp() const; // exponential function
private:
    double rpart; // real part
    double ipart; // imaginary part
};

```

Here is a run of my program. Note that my program reads $a + bi$ by reading the number a followed by the number b .

```
./a.out
Enter a complex number z: 0 -1
z = -i !z = i z-3 = -3-i 2*z = -2i 2+z = 2-i 3/z = 3i z^2 = -1
Enter a complex number w: 1 1
w = 1+i z+w = 1 z-w = -1-2i z*w = 1-i z/w = -0.5-0.5i
Enter a complex number z: 4 3
z = 4+3i !z = 4-3i z-3 = 1+3i 2*z = 8+6i 2+z = 6+3i 3/z = 0.48-0.36i z^2 = 7+24i
Enter a complex number w: 1 2
w = 1+2i z+w = 5+5i z-w = 3+i z*w = -2+11i z/w = 2-i
Enter a complex number z: 6 3
z = 6+3i !z = 6-3i z-3 = 3+3i 2*z = 12+6i 2+z = 8+3i 3/z = 0.4-0.2i z^2 = 27+36i
Enter a complex number w: 0 0
a.out: complex.cpp:248: cmplx cmplx::inverse() const: Assertion '*this!=0' failed.
w = 0 z+w = 6+3i z-w = 6+3iAborted
```

z/w could not be computed because division by zero is undefined. My program computes z/w by multiplying z by the inverse of w . Since $w = 0$, w^{-1} is undefined.