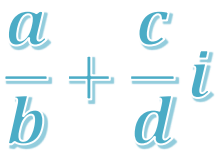

CS132 – PROGRAMMING ASSIGNMENT

FRACTIONS

OVERVIEW

This activity is designed to have you work with a class and practice adding field methods and operator overloading.

INSTRUCTIONS

In this program you will write a class that is designed to combine the concepts of integer fractions and complex numbers. Remember from math class that a fraction is number written in the form $\frac{a}{b}$ where a and b are not reducible. $\frac{1}{2}$ is acceptable whereas $\frac{2}{4}$ is not because it can be reduced by a power of two.

However we are also going to augment that with the idea of a complex number. Remember that the complex numbers are a joint number consisting of a real part (a) and an imaginary part (b * i) where i is the imaginary number $\sqrt{-1}$.

Your class will implement the FractionalComplex number class, and allow for the class to implement the necessary overloaded functions and operators.

You will be provided a main program that is designed to test your class, and make sure that each part is working as indicated.

Note that if you need help with the math involved, tips will be provided at the end, but feel free to get help from a math instructor or from your computer science instructor.

THE MAIN PROGRAM.

Your main program will be provided for you. It will be broken into pieces that allow you to get parts of your program up and running in small sections without having to get the entire thing working at once.

You are not to modify the main program code as a norm, but sections of it are commented out. It will be important for you to uncomment these sections as you get parts of your code working.

THE FRACTIONALCOMPLEX CLASS

The primary work that you will be doing is creating the `FractionalComplex` class and its associated methods. You will be expected to implement all of the methods below, but as always, you are welcome to add additional private methods if you want / need them to help you out.

Your class should only use int values, as the point of the class is to do things with fractions.

A Fractional Complex number will be represented by 4 integer values, a, b, c, and d. The format of the number will be

$$\frac{a}{b}+\frac{c}{d}i$$

Note that you should not have to keep track of the *(i)* as that will be in the code. At all times $\frac{a}{b}$ should be kept in reduced form as should $\frac{c}{d}$.

$\frac{1}{2}+\frac{4}{5}i$ is acceptable, but $\frac{5}{10}+\frac{4}{5}i$ is not because $\frac{a}{b}$ is not reduced.

If at any time one of the numbers is zero, represent it by the number $\frac{0}{1}$. So $\frac{0}{1}+\frac{0}{1}i$ is acceptable.

Make sure that the denominators (b,d) are never zero. If they become zero in the constructor or by work, then make the value $\frac{0}{1}$ by default.

Note that these are fractions. Make sure you add/subtract/multiply/divide according the rules for fractions.

If a fraction is negative, then the negative number should be in the numerator. If they are both negative then make it positive. $\frac{1}{2}$ and $\frac{-1}{2}$ are ok. $\frac{1}{-2}$, $\frac{-1}{-2}$ are not ok.

PHASE ONE

FRACTIONALCOMPLEX ()

You will want a default constructor that creates the number $\frac{0}{1}+\frac{0}{1}i$.

FRACTIONALCOMPLEX (INT X, INT Y)

You will want a two variable constructor that creates the number $\frac{x}{1}+\frac{y}{1}i$.

FRACTIONALCOMPLEX (INT A, INT B, INT C, INT D)

You will want a four variable constructor that creates the number $\frac{a}{b}+\frac{c}{d}i$. Make sure to watch for invalid numbers and reduce after creation.

PRIVATE REDUCE() – *OPTIONAL BUT SUGGESTED*

You might want to add a private method that double checks both a/b and c/d to make sure that they are reduced. You might be using this quite often, so make sure that it is a method. This is also where you should check the positive/negative of each fraction.

PUBLIC PRINTME()

Eventually we will be overloading the << method, but for initial testing, create a `printme()` method that outputs the `FractionalComplex` in the form.

- `[(a/b) + (c/d)i]`

So if a = 2, b = 5, c = 10, d = 11. Then the output of `printme()` should look like:

- `[(2/5) + (10/11)i]`

PHASE 2

OPERATOR+

Create an operator that allows you to add two `FractionalComplex` together to create a third one. Adding complex numbers requires you to add the real parts together and the imaginary parts together.

So if $x=\frac{1}{2}+\frac{3}{4}i$ and $y=\frac{1}{3}+\frac{5}{4}i$ then the code

- `z = x + y`

Should make $z = \frac{5}{6} + \frac{2}{1}i$. Note that he numbers were added then simplified. $\frac{1}{2} + \frac{1}{3} = \frac{5}{6}$ and $\frac{3}{4} + \frac{5}{4} = \frac{8}{4} = \frac{2}{1}$

OPERATOR-

Create an operator that allows you to subtract two `FractionalComplex` together to create a third one. Subtracting complex numbers requires you to subtract the real parts together and the imaginary parts together.

$$\left[\frac{1}{2} + \frac{4}{5}i\right] - \left[\frac{1}{3} + \frac{2}{3}i\right] = \left[\frac{1}{6} + \frac{2}{15}i\right]$$

PHASE 3

DOUBLE LENGTH()

The length of a complex number is the value of the real part squared + the imaginary part squared square rooted.

The length of $\left[\frac{1}{2} + \frac{4}{5}i\right]$ is $\sqrt{(.5)^2 + (.8)^2} = .943398$

Note that this is probably the only part of the class that should use decimal numbers.

OPERATOR==

Create an operator that allows you to check to see if two `FractionalComplex` numbers are equal to each other. To be equal to each other, all values of a,b,c and d must match.

OPERATOR<

Create an operator that allows you to check to see if one `FractionalComplex` is smaller than another. To compare two fractional numbers, you check to see if one is longer than the other using the `length()` method;

OPERATOR>

Create an operator that allows you to check to see if one `FractionalComplex` is larger than another. To compare two fractional numbers, you check to see if one is longer than the other using the `length()` method;.

PHASE 4

OPERATOR <<

Create an overloaded member function that allows you to cout/stream a `FractionalComplex` using the standard notation.

```
cout << fc1 << " " << fc2 << " " <<fc3 <<endl;
```

Should work. The output should be the same as `printme()` above, but now it can be used in the traditional stream notation.

PHASE 5

OPERATOR*(FRACTIONALCOMPLEX)

Create an overloaded member function that allows you to multiply a `FractionalComplex` by another `FractionalComplex` .

Note that multiplying Fractional complex numbers is tricky.

$$\left[\frac{a}{b}+\frac{c}{d}i\right]\cdot\left[\frac{w}{x}+\frac{y}{z}i\right]=\left[\left(\frac{aw}{bx}+\frac{ay}{bz}i\right)+\left(\frac{-1cy}{dz}+\frac{cw}{dx}i\right)\right]$$

Then you can use the addition method on the two fractions on the inside.

So you should be able to type

- `z = x * y`

OPERATOR*(INT)

Create an overloaded member function that allows you to multiply a `FractionalComplex` by an int.

Note that multiplying Fractional complex numbers by an integer is easy.

$$\left[\frac{a}{b}+\frac{c}{d}i\right]\cdot x=\left[\frac{ax}{b}+\frac{cx}{d}i\right]$$

Then reduce.

So you should be able to type :

- `z = x * 7;`

PHASE 5

OPERATOR ++ (TWO OF THEM)

Create an overloaded member function that allows you pre-increment and one that allows you to post-increment a `FractionalComplex` .

When you pre/post increment a `FractionalComplex` you should add 1 to both the real and the imaginary part.

$$\left[\frac{1}{2}+\frac{3}{4}i\right]++=\left[\frac{3}{2}+\frac{7}{4}i\right]$$

Make sure that both `fc++` and `++fc` works.

SUBMISSION

You will be expected to turn in 2 files

- FractionalComplex .h
- FractionalComplex .cpp

Before submitting your assignment, include a comment at the beginning of your program with some basic information and a description of the program in your own words. For example:

```
// Suzy Student, CS132, Autumn 2049, Section XX
// Programming Assignment #2, 06/07/49
//
// This program's behavior is ...
```

Make sure to comment appropriately and document all your major points.

PROGRAM NOTES

Get help from someone including me if you need help with the math. The purpose of this is to implement the math, not spend forever on the math itself.

Do one thing at a time, don't try to program the entire thing in one piece.

Remember to add fractions you have to have a common denominator, but if you can't find one, you can do the following.

$\frac{a}{b} + \frac{c}{d} = \frac{ad+cb}{bd}$ then reduce.

To reduce a fraction without knowing the numbers, remember you are on a computer, there are ways to brute force it that are fast for a computer that humans shouldn't do.