Homework 3 (25 points)

cs3157 – Advanced Programming
Prof. Shlomo Hershkop
Dept of Computer Science
Columbia University
Spring 2006
Due: May 7 11:59 pm.

Objective:

1. Practice your cpp (classes, overriding operators, and ~~templates~~), html, and cgi skills
2. do something interesting with CPP classes

You will be building a fraction class in cpp to represent fractional numbers

What this means is that you add 1/2 and 1/2, you'll get 1, but if you multiply 1/4 and 1/4, you'll get 1/16 (not 0.0625).

In addition, you have to override several arithmetic and I/O functions in order to make your class elegant to use. You'll be writing several member functions, but most of them will be small. (There's no dynamic memory here, yay!.) The majority of your code--the implementation of your class--will be in fraction.cpp. The class declarations will be in fraction.h, and a test program (for evaluating the test expressions) will be testFraction.cpp. You will also need a README and makefile.

To make the assignment that much easier, I will outline some simple steps to help you get everything done quickly

1) Create the basic fraction class
    a. You can keep two number, the numerator and denominator
    b. Create a simple constructor which takes 2 ints
        i. Make sure only the numerator is negative if applicable, that is 3/-5 => -3/5, -4/-5 = 4/5
    c. Create a destructor (even if no work done here)
    d. Overload the << so you can print out the fraction using cout
        i. Remember that if the denominator is a one, you only print the numerator 45/1 should be printed as 45
        ii. A zero anywhere should be printed as 0 so that 0/23 and 23/0 is 0
    e. Create the following code in the test class:

```
Fraction A = Fraction(3,4);
Fraction B = Fraction (5,9);
cout<<"Fraction test:"<<endl
cout<<"A is "<<A << " B is "<<B<<endl
```

2) Create a reduce function – create a function which takes a Fraction and returns an equivalent one reduced, that is given 4/20 it return 1/5
Code:
here some code which will return the GCD between 2 numbers, make sure you are only passing in by value ☺ use it in your reduce code ☺

```
int GCD(int num, int denom) {
     // while num and denom are not equal
     while (num != denom) {
     // modify a and b until they become equal
     if (num>denom)
          num = num-denom;
     else if (num<denom)
          denom = denom-num;
     }
// now num == denom, and both represent result
return num;
}
```

3) Overloading operaters
   a. Overload the + operator
   b. Overload the – operator
   c. Overload the * operator
   d. Overload the / (division) operator, remember you need to invert then multiply
   e. Overload the += operator
   f. Overload the == operator
   g. Create an assignment constructor
   h. Add the following to your test file:

```
Fraction C = A + B;
Fraction D = A / B ;
Fraction E = D * Fraction(-1,1);
Fraction F = Fraction(12 , 49) * Fraction(34, 5);
cout << “C is “ << C << “ and D is “ <<D <<endl;
cout << “Negation of D is “ << E << endl;
cout <<”F is “ << F << endl;
```

4) Now think about how to add templates….hum while you enjoy not having to debug it ☺
5) If you are not clear about a step, either document your assumptions in your README or contact me for clarification
6) Extra Credit:
   create a CGI front-end so that one can either enter in 2 fractions and choose a specific operation (+, -, /, *) and see the results on the screen or enter a single fraction and see it reduced…..

Start early and Good Luck (you will need it if you don’t start early) !!