CS 2A Object-Oriented Programming in C++

Assignment 8

Assignment 8.1 [75 points]

Note: No documentation is required on this assignment

Write a fraction class whose objects will represent fractions. For this assignment you aren't required to reduce your fractions. You should provide the following member functions:

1. A set() operation that takes two integer arguments, a numerator and a denominator, and sets the calling object accordingly.

2. Arithmetic operations that add, subtract, multiply, and divide fractions. These should be implemented as value returning functions that return a fraction object. They should be named addedTo, subtract, multipliedBy, and dividedBy. In these functions you will need to declare a local "fraction" variable, assign to it the result of the mathematical operation, and then return it.

3. A boolean operation named isEqualTo that compares two fraction objects for equality. Since you aren't

reducing your fractions, you'll need to do this by cross-multiplying. A little review: if numerator1 * denominator2 equals denominator1 * numerator2, then the fractions are equal.

4. An output operation named print that displays the value of a fraction object on the screen in the form numerator/denominator.

Your class should have exactly two data members, one to represent the numerator of the fraction being represented, and one to represent the denominator of the fraction being represented.

Here's a hint for how you will set up your arithmetic operation functions: You need two fractions. One is the parameter, one is the calling object. The function multiplies the calling object times the parameter and returns the result. In some ways it is similar to the comesBefore() function from the lesson. That function also needs two fractions, and one is the calling object and one is the parameter.

When adding or subtracting fractions, remember that you must first find the common denominator. The easy way to do this is to multiply the denominators together and use that product as the common denominator.

I am providing a client program for you below. You should copy and paste this and use it as your client program. The output that should be produced when the provided client program is run with your class is also given below, so that you can check your results.

I strongly suggest that you design your class incrementally. For example, you should first implement only the set function and the output function, and then test what you have so far. Once this code has been thoroughly debugged, you should add additional member functions, testing each one thoroughly as it is added. You might do this by creating your own client program to test the code at each stage; however, it would probably be better to use the provided client program and comment out code that relates to member functions that you have not yet implemented.

As you can see from the sample output given below, you are not required to reduce fractions or change improper fractions into mixed numbers for printing. Just print it as an improper fraction. You are also not required to deal with negative numbers, either in the numerator or the denominator.

Here is the client program.

```
#include <iostream>
using namespace std;

int main()
{
    fraction f1;
    fraction f2;
    fraction result;

    f1.set(9, 8);
    f2.set(2, 3);

    cout << "The product of ";
    f1.print();
    cout << " and ";
    f2.print();
    cout << " is ";
    result = f1.multipliedBy(f2);
    result.print();
    cout << endl;

    cout << "The quotient of ";
    f1.print();
    cout << " and ";
    f2.print();
    cout << " is ";
    result = f1.dividedBy(f2);
    result.print();
    cout << endl;

    cout << "The sum of ";
    f1.print();
    cout << " and ";
    f2.print();
    cout << " is ";
    result = f1.addedTo(f2);
    result.print();
    cout << endl;

    cout << "The difference of ";
    f1.print();
    cout << " and ";
    f2.print();
    cout << " is ";
    result = f1.subtract(f2);
    result.print();
    cout << endl;

    if (f1.isEqualTo(f2)){
        cout << "The two fractions are equal." << endl;
    } else {
        cout << "The two fractions are not equal." << endl;
    }
}
```

This client should produce the output shown here:

```
The product of 9/8 and 2/3 is 18/24
The quotient of 9/8 and 2/3 is 27/16
The sum of 9/8 and 2/3 is 43/24
The difference of 9/8 and 2/3 is 11/24
The two fractions are not equal.
```

You may not change the client program in any way. Changing the client program will result in a grade of 0 on the project.

Note: in future weeks you will be submitting the client program and class in separate files, but for this week you will be putting all of your code, class and client program, into a single file to submit it. The class declaration will come first, followed by the definitions of the class member functions, followed by the client program.

Submit Your Work

Name your source code file(s) according to the assignment number (a1_1.cpp, a4_2.cpp, etc.). Execute each program and copy/paste the output into the bottom of the corresponding source code file, making it into a comment. Use the Assignment Submission link to submit the source file(s). When you submit your assignment there will be a text field in which you can add a note to me (called a "comment", but don't confuse it with a C++ comment). In this "comments" section of the submission page let me know whether the program(s) work as required.