Project 9

Objective:

Develop a C++ application to manage fractions using an Abstract Data Type (ADT).

Project Files:

fraction.h: This header file defines the interface for the Fraction class.

fraction.cpp: Contains the implementation of the Fraction class.

FractionTester.cpp: A C++ file used to test and demonstrate the functionality of the Fraction class. This file should contain the main function.

Fraction Basics:

A fraction is defined as p/q where p is the numerator, and q is a non-zero denominator. Numbers like 3/4, 15/-20.

Fraction Class Details:

Member Variables:

numerator: an integer representing the top part of the fraction.

denominator: an integer representing the bottom part of the fraction (cannot be zero).

Constructors:

Two-parameter Constructor: Initializes the fraction with specified numerator and denominator, ensuring the denominator is not zero.

Single-parameter Constructor: Treats the number as a whole number with the denominator set to 1.

Default Constructor: Initializes the fraction to represent zero (0/1).

Operator Overloading:

Input (>>) and output (<<) operators to handle fraction formats such as 1/2, 15/32, or −300/−401.

Arithmetic and comparison operators: ==, <, <=, >, >=, +, -, and /.

So, the program must have ten operator overloading functions.

Mathematical Operations:

Equality and Comparison: Uses cross multiplication to evaluate equality and comparisons.

Normalization: Adjusts fractions so the denominator is always positive, and the fraction is reduced to its simplest form.

Testing with FractionTester.cpp:

Enhanced User Interaction:

In the main function, implement a loop to continuously prompt the user to enter fractions until they choose to stop. Store all user-generated fractions in a dynamic array, which means using a pointer to manage an array that can grow as needed

Fraction Selection and Operations: After entering fractions, allow the user to select two fractions from the array by specifying their indices. Perform all the mentioned operations (arithmetic and comparison) on these two selected fractions and display the results.

Submission Guidelines:

Please submit the following files:

fraction.h

fraction.cpp

FractionTester.cpp

Ensure your program accurately implements and demonstrates all specified functions. Compilation errors will result in a grade of zero. Ensure rigorous testing to avoid any issues.

Sample output:

```
Enter a fraction (numerator/denominator): 1/2
Do you want to enter another fraction? (y/n): y
Enter a fraction (numerator/denominator): 2/-8
Do you want to enter another fraction? (y/n): y
Enter a fraction (numerator/denominator): -3/9
Do you want to enter another fraction? (y/n): y
Enter a fraction (numerator/denominator): 2/5
Do you want to enter another fraction? (y/n): n
Select the index of the first fraction: 1
Select the index of the second fraction: 2
First fraction: -1/4
Second fraction: -1/3
Sum: -7/12
Difference: 1/12
Product: 1/12
Quotient: 3/4
```