Kahlil Bello

Project 10

Nov 28, 2018

**HEADER FILE**

#pragma once

#ifndef FRACTION

#define FRACTION

#include <iostream>

#include <algorithm>

#include <iomanip>

#include <string>

using namespace std;

class fract

{

private:

int numerator, denominator;

public:

void readFraction()

{

cout << "Enter a fraction:"; //ex 2/3

cin >> numerator;

cin.ignore();

cin >> denominator;

}

void display()

{

cout << numerator << "/" << denominator << " ";

}

void plusOne()

{

cout << "(" << numerator << "/" << denominator << ")++ = " << (numerator + 1) << "/" << (denominator + 1) << endl;

}

void minusOne()

{

cout <<"("<<numerator<<"/"<<denominator<<")-- = "<< (numerator - 1) << "/" << (denominator - 1) << endl;

}

friend fract operator+(fract f1, fract f2);

friend fract operator-(fract f1, fract f2);

friend fract operator\*(fract f1, fract f2);

friend fract operator/(fract f1, fract f2);

};

fract operator+(fract f1, fract f2)

{

fract res;

res.numerator = (((f1.numerator) \* (f2.denominator)) + ((f1.denominator) \* (f2.numerator)));

res.denominator = (f1.denominator \* f2.denominator);

return res;

}

fract operator-(fract f1, fract f2)

{

fract res;

res.numerator = (((f1.numerator) \* (f2.denominator)) - ((f1.denominator) \* (f2.numerator)));

res.denominator = (f1.denominator \* f2.denominator);

return res;

}

fract operator\*(fract f1, fract f2)

{

fract res;

res.numerator = ((f1.numerator) \* (f2.numerator));

res.denominator = (f1.denominator \* f2.denominator);

return res;

}

fract operator/(fract f1, fract f2)

{

fract res;

res.numerator = ((f1.numerator) \* (f2.denominator));

res.denominator = (f1.denominator \* f2.numerator);

return res;

}

#endif

**MAIN/SOURCE.CPP**

#include "FRACTION.h"

int main()

{

fract f1, f2, f3;

f1.readFraction();

f2.readFraction();

f3 = f1 + f2;

f1.display(); cout << "+ ";

f2.display(); cout << "=";

f3.display(); cout << endl;

f3 = f1 - f2;

f1.display(); cout << "- ";

f2.display(); cout << "=";

f3.display(); cout << endl;

f3 = f1 \* f2;

f1.display(); cout << "\* ";

f2.display(); cout << "=";

f3.display(); cout << endl;

f3 = f1 / f2;

f1.display(); cout << "/ ";

f2.display(); cout << "=";

f3.display(); cout << endl;

f2.plusOne();

f1.minusOne();

cout << endl;

system("pause");

return 0;

}

**OUTPUT**

