//Joon Im

//Marie Payad

//Demo:9:11

#include "Fraction.h"

#include <iostream>

using namespace std;

int main()

{

Fraction f1(1,3), f2(1,6), f3;

f3 = f1.add(f2);

cout << "Adding: (1/2) + (1/6)= " << endl;

f3.print();

cout << "Multiplying: " << endl;

Fraction f4(1,2), f5(2,3), f6;

f6 = f4.mul(f5);

cout << "1/2 \* 2/3 = \n";

f6.print();

return 0;

}

#include <iostream>

#include <cmath>

#include <cassert>

#include "Fraction.h" denom = den;

}

normalize ();

}

Fraction::Fraction()

: numer (0), denom (1)

{

}

Fraction::Fraction (const Fraction& fract)

: numer (fract.numer), denom (fract.denom)

{

}

Fraction::~Fraction ()

{

}

int Fraction::getNumer() const

{

return numer;

}

int Fraction::getDenom() const

{

return denom;

}

void Fraction::print() const

{

cout << numer << "/" << denom << endl;

}

void Fraction::setNumer (int num)

{

numer = num;

normalize();

}

void Fraction::setDenom (int den)

{

denom = den;

normalize();

}

void Fraction::normalize()

{

// Handling a denominator of zero

if (denom == 0)

{

cout << "Invalid denomination. Need to quit." << endl;

assert (false);

}

// Changing the sign of denominator

if (denom < 0)

{

denom = - denom;

numer = - numer;

}

// Dividing numerator and denominator by gcd

int divisor = gcd (abs(numer), abs (denom));

numer = numer / divisor;

denom = denom / divisor;

}

int Fraction :: gcd (int n, int m)

{

int gcd = 1;

for (int k = 1; k <= n && k <= m; k++)

{

if (n % k == 0 && m % k == 0)

{

gcd = k;

}

}

return gcd;

}

Fraction Fraction::add(Fraction &f)

{

int numSum = (numer \* f.denom) + (denom \* f.numer);

int denSum = (denom \* f.denom);

int divisor = gcd(numSum, denSum);

numSum = numSum / divisor;

denSum = denSum / divisor;

Fraction sum(numSum, denSum);

return sum;

}

Fraction Fraction::mul(Fraction &f)

{

int numSum = (numer \* f.numer);

int denSum = (denom \* f.denom);

int divisor = gcd(numSum, denSum);

numSum = numSum / divisor;

denSum = denSum / divisor;

Fraction sum(numSum, denSum);

return sum;

}

#include <iostream>

using namespace std;

class Fraction

{

// Data members

private:

int numer;

int denom;

// Public member functions

public: // Constructors

Fraction (int num, int den);

Fraction ();

Fraction (const Fraction& fract);

~Fraction ();

Fraction add(Fraction &f);

Fraction mul (Fraction &f);

// Accessors

int getNumer () const;

int getDenom () const;

void print () const;

// Mutators

void setNumer (int num);

void setDenom (int den);

// Helping private member functions

private:

void normalize ();

int gcd (int n, int m);

};

