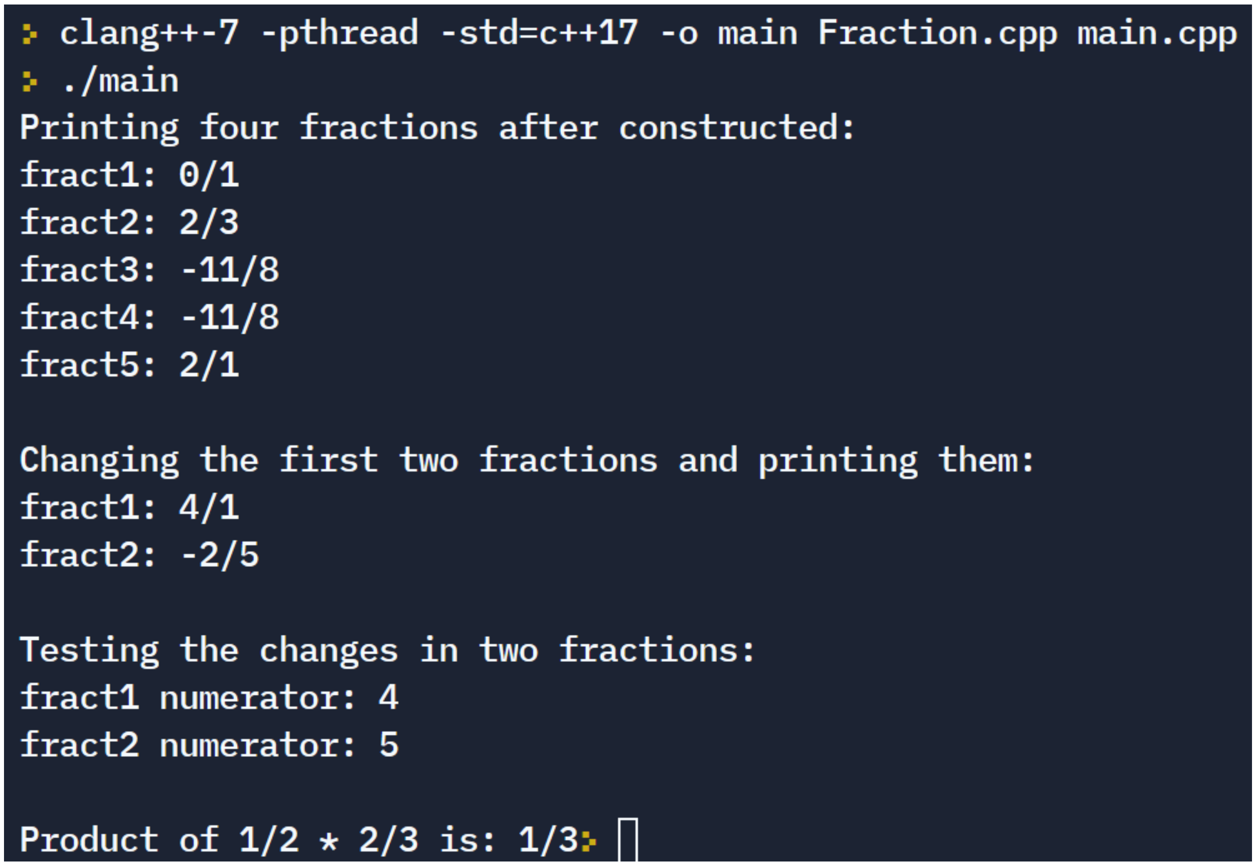
Jeremy Howell, Nhi Pham

CECS 282 – Sec 07

9/7/2021

Lab 3-2: Fractions



```
› clang++-7 -pthread -std=c++17 -o main Fraction.cpp main.cpp
› ./main
Printing four fractions after constructed:
fract1: 0/1
fract2: 2/3
fract3: -11/8
fract4: -11/8
fract5: 2/1

Changing the first two fractions and printing them:
fract1: 4/1
fract2: -2/5

Testing the changes in two fractions:
fract1 numerator: 4
fract2 numerator: 5

Product of 1/2 * 2/3 is: 1/3›
```

Demonstrated at 11:18 am on September, 7th 2021

Fraction.h

```
#ifndef FRACTION_H
#define FRACTION_H

#include <iostream>

using namespace std;

class Fraction {
  private:
    int numer;
    int denom;
  public:
    Fraction(int num, int den);
    Fraction();
    Fraction(const Fraction& fract);
    ~Fraction();

    int getNumer() const;
    int getDenom() const;
    void print() const;

    void setNumer(int num);
    void setDenom(int den);

  private:
    void normalize();
    int gcd(int n, int m);
};

#endif
```

Fraction.cpp

```
#include <iostream>
#include <cmath>
#include <cassert>
#include "Fraction.h"

using namespace std;

Fraction::Fraction(int num, int den) {
  if (den == 0){
    den = 1;
  }
  numer = num;
  denom = den;
  normalize();
}

Fraction::Fraction() {
  numer = 0;
  denom = 1;
}
```

```
Fraction::Fraction(const Fraction& fract){
  numer = fract.getNumer();
  denom = fract.getDenom();
}

Fraction::~Fraction() {

}

int Fraction::getNumer() const {
  return numer;
}

int Fraction::getDenom() const {
  return denom;
}

void Fraction::print() const {
  cout << numer << "/" << denom;
}

void Fraction::setNumer(int num) {
  numer = num;
  normalize();
}

void Fraction::setDenom(int den) {
  denom = den;
  normalize();
}

void Fraction::normalize() {
  if (denom == 0) {
    cout << "Invalid denomination. Need to quit." << endl;
    assert(false);
  }

  if (denom < 0) {
    denom = -denom;
    numer = -numer;
  }

  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;
}
```

main.cpp

```
#include <iostream>
#include "Fraction.h"

using namespace std;

Fraction result;
Fraction& multiplyFract(Fraction& fr1, Fraction& fr2) {
  result.setNumer(fr1.getNumer() * fr2.getNumer());
  result.setDenom(fr1.getDenom() * fr2.getDenom());

  return result;
}

int main() {
  Fraction fract1;
  Fraction fract2(14, 21);
  Fraction fract3(11, -8);
  Fraction fract4(fract3);
  Fraction fract5(2, 0);

  cout << "Printing four fractions after constructed: " << endl;
  cout << "fract1: ";
  fract1.print();
  cout << endl;
  cout << "fract2: ";
  fract2.print();
  cout << endl;
  cout << "fract3: ";
  fract3.print();
  cout << endl;
  cout << "fract4: ";
  fract4.print();
  cout << endl;
  cout << "fract5: ";
  fract5.print();
  cout << endl;
  cout << endl;

  cout << "Changing the first two fractions and printing them: ";
  cout << endl;
  fract1.setNumer(4);
  cout << "fract1: ";
  fract1.print();
  cout << endl;
  fract2.setDenom(-5);
  cout << "fract2: ";
  fract2.print();
  cout << endl;
  cout << endl;

  cout << "Testing the changes in two fractions: " << endl;
  cout << "fract1 numerator: " << fract1.getNumer() << endl;
  cout << "fract2 numerator: " << fract2.getDenom() << endl;
  cout << endl;

  fract1.setNumer(1);
  fract1.setDenom(2);
  fract2.setNumer(2);
  fract2.setDenom(3);

  Fraction fract6 = multiplyFract(fract1, fract2);
  cout << "Product of " << fract1.getNumer() << "/" << fract1.getDenom() << " * "
<< fract2.getNumer() << "/" << fract2.getDenom() << " is: ";
  fract6.print();

  return 0;
}
```