

CS 215 – Spring 2021

Lab 6: Fraction FUNctions

Learning Objective:

- Coding and testing of functions given a detailed design

General Description:

You are to write and test a set of functions dealing with a Fraction, based on detailed function designs. You will also write a main program to allow the user to test the functions.

Detailed Design Specifications:

Use the following structure for a fraction:

```
struct frac {    // a fraction structure
    int num; // numerator
    int den; // denominator
};
```

1. Write a function called **AskFraction** which asks the user to enter a numerator and a denominator. The function should assume the user enters integers, but should validate the denominator is a positive integer (the numerator may be negative or 0) It should repeatedly ask the user to re-enter the denominator when they enter a denominator of 0 or less. The function returns the fraction entered. A function can return an object...just declare a local object, populate it, and return it. Example:

```
struct thing { string name; float price; };

thing InitNewThing() {
    thing f;           // declare a local object, not an argument
    f.name = "";
    f.price = 0.0;
    return f;          // return local object
}
```

2. Write a function called **DecimalValue** which is given a fraction, then calculates and returns the decimal value of the fraction (ex: $\frac{1}{2}$ results in 0.5).
3. Write a function called **PrintFraction** which is given a fraction. It prints 3 lines:
 - o the numerator (use a width of 3)
 - o four dashes, an equal sign, and the decimal value. Invoke `DecimalValue()` to do this.
 - o the denominator (use a width of 3)
4. Write a function called **Multiply** that is given two fractions and returns the product of the given fractions. Remember: $A/B * C/D = (A*C) / (B*D)$

5. Write a function called **Add** that is given two fractions and returns the sum of the two fractions. When the denominators are the same, just add the numerators:

$$1/5 + 2/5 = 3/5$$

When the denominators differ, do a cross product to get the same denominator for each:

$$2/3 + 4/5 = (2*5) + (4*3) / (3*5) = 22/15$$

6. Write a function called **Simplify** that *modifies* a fraction (not given/returns...it is void) by simplifying it. To simplify, divide both the numerator and denominator by every integer between 2 and the denominator (using a for loop), counting backwards. If both numerator and denominator are *evenly divisible* by one such integer, then divide both by that number. (there are better ways, if you know how, feel free to use them WITHOUT using any C++ library functions). Example:

Given { num=60, den=90 } so I = 60 to 2

num	den	i	are both evenly divisible by i?
60	90	60	no
60	90	59	no
60	90	58..18	no
60	90	17	no
60	90	16	no
60	90	15	YES – divide numerator and denominator by 15
4	6	14	no
4	6	13..3	no
4	6	2	YES
2	3		final answer: 2/3

Hint: use C++'s int division to test for "is evenly divisible"... $(A/B)*B == A$

$$\begin{aligned} (5/2)*2 &= 2*2 = 4 \neq 5 && \text{(so 5 is not div by 2)} \\ (6/2)*2 &= 3*2 = 6 == 6 && \text{(so 6 is div by 2)} \end{aligned}$$

The main() is written for you. See the given starting code on the course website.

No functions should use cin or cout except:

- PrintFraction() uses couts to print the values
- AskFraction() uses cout(s) and cin(s) to ask the user for a numerator and denominator

Style:

Add a comment box for each function. (This is in addition to the regular comment box at the top)

Example:

```
//-----
//                               PrintFraction
//-----
```

Sample Executions:

Denominator validated.

Fractions have different denominators.

Simplified product same as unsimplified.

```
FRACTION 1:
Enter numerator:      2
Enter denominator > 0: 0
Enter denominator > 0: -3
Enter denominator > 0: -4
Enter denominator > 0: 5
```

```
FRACTION 2:
Enter numerator:      1
Enter denominator > 0: 3
PRODUCT:
```

```
  2
---- = 0.133333
 15
```

```
SUM:
 11
---- = 0.733333
 15
```

```
PRODUCT SIMPLIFIED:
```

```
  2
---- = 0.133333
 15
```

Press any key to continue . . . _

Fractions have different denominators.

Simplified product differs from original.

```
FRACTION 1:
Enter numerator:      1
Enter denominator > 0: 5
```

```
FRACTION 2:
Enter numerator:      5
Enter denominator > 0: 7
PRODUCT:
```

```
  5
---- = 0.142857
 35
```

```
SUM:
 32
---- = 0.914286
 35
```

```
PRODUCT SIMPLIFIED:
```

```
  1
---- = 0.142857
  7
```

Press any key to continue . . . _

Fractions have the same denominator:

```
FRACTION 1:
Enter numerator:      1
Enter denominator > 0: 5
```

```
FRACTION 2:
Enter numerator:      2
Enter denominator > 0: 5
PRODUCT:
```

```
  2
---- = 0.08
 25
```

```
SUM:
  3
---- = 0.6
  5
```

```
PRODUCT SIMPLIFIED:
```

```
  2
---- = 0.08
 25
```

Press any key to continue . . . _

Grading Rubrics:

Style:

- comment box at top with Lab 6, Name and Section
- comment box for each function, including main(), with name centered
- use of good identifier names
- proper spacing/indentation

Code:

- each function is written as specified with the correct arguments and return types
- no couts in any function except PrintFraction() and AskFraction()
- no cins in any function except AskFraction()
- user input for denominator is validated
- fractions are printed in the correct format
- results (product, sum and simplified product) are correct