Search entries or author

This is a graded discussion: 5 points possible

due Sep 18

92 93

Assignment 4 Discussion

Post your contribution to the assignment 4 discussion here. This could involve asking a question, answering another student's question, giving an example of something that you struggled with and then overcame (or didn't!), giving an example of something you found particularly cool, or any other constructive way you can think of to participate.

Reply

○ (https://santarosajc.instructure.com/courses/24402/users/74745)

James O'Hara

(https://santarosajc.instructure.com/courses/24402/users/74745)

Sep 9, 2017

Jim's Hopefully Helpful Hints for Assignment 4.

When designing the extraction overload and the insertion overload enhancement, be sure to walk through some test cases with positive and negative proper fractions, improper fractions, mixed numbers. The input file doesn't have negative denominators.
Be sure to check your output against the correct output given in the Assignment. For the extraction overload, the most frequent errors are on the last two records of the input file. Be sure to check your output carefully for those two.

Don't reduce fractions in the insertion overload. It will mask unreduced fractions resulting from failing to call simplify() in the correct places. I'll be watching for that :)

In simplify(), should we start with a large trial divisor and work down, or a small trial divisor and work up? Both will work, but which will work better? Faster? Should the numerator or the denominator be the trial divisor? The product of the numerator and the denominator? The larger or smaller of the two? Is -20 larger than +10? Will your code pick the value you intend to pick?

Should simplify() be void, or should it return a value? Should it take parameters? Do ++ or -- need to call simplify()?

Should you assert that a denominator argument is not zero? Or substitute a valid value? Why or why not?

Be sure to study Style Conventions 1D.1 and 1D.2. Study Style Convention 7p and how it relates to the relational operator overloads.

(https://santarosajc.instructure.com/courses/24402/users/59022)

Drew Rodrigues

○ (https://santarosajc.instructure.com/courses/24402/users/59022)

Search entries or author

"In sin

The way I did it was started with the lowest number:

Since when simplifying, the largest number that divides into both numbers can't be greater than the smaller of the two. It reduces the amount of iterations.

I also made my simplify() method void, since it's just working directly on the object as a private method, you don't need to return anything, just work with "this".

"Should you assert that a denominator argument is not zero? Or substitute a valid value? Why or why not?"

The teacher mentioned to assert that the denominator isn't zero, which I placed in my constructor/initializer function.

assert(inDenominator != 0);

I hope that helps a bit!

*Edited by **David Harden (https://santarosajc.instructure.com/courses/24402/users/60154)** on Sep 16 at 3:04pm*

↰ **Reply**

○ **(https://santarosajc.instructure.com/courses/24402/users/114750)** Terrence Dubois

(https://santarosajc.instructure.com/courses/24402/users/114750)

Sep 14, 2017

For my simplify algorithm, I first look for the higher value between the numerator and denominator. I set the GCF variable to that value. From there, I loop downwards checking to see if both the numerator and denominator give a remainder of zero when divided by the GCF variable. If they do, then that is the GCF. If not, decrement the GCF variable and try again. I don't think it's the most efficient way, but luckily the instructions say to not worry about that!

On a side note: Where is everybody? It's already Thursday!

○ **(https://santarosajc.instructure.com/courses/24402/users/59022)** Drew Rodrigues

(https://santarosajc.instructure.com/courses/24402/users/59022)

Friday

To make it more efficient, you should begin with the lower number of the numerator and denominator. Since once dividing, numbers greater than a number are not factors of that number. For instance:

400/2

Search entries or author

If you r

of the g

by 1 is pointless since it'll always give a remainder of zero with integers. That also removes 1 iteration from the loop.

Hope this helps a bit!

Kyle Stewart (https://santarosajc.instructure.com/courses/24402/users/85990)

(https://santarosajc.instructure.com/courses/24402/users/85990)

Saturday

Adding on to what Drew said, don't begin with a negative number. If you have -4/8, you want to start with 4 and not -4.

Noah Bartel (https://santarosajc.instructure.com/courses/24402/users/99345)

(https://santarosajc.instructure.com/courses/24402/users/99345)

Monday

So just comparing the absolute value would work here then right?

David Harden (https://santarosajc.instructure.com/courses/24402/users/60154)

(https://santarosajc.instructure.com/courses/24402/users/60154)

Monday

Yes.

Reply

(https://santarosajc.instructure.com/courses/24402/users/114708) Jesse Browne

(https://santarosajc.instructure.com/courses/24402/users/114708)

Friday

I have a couple questions.

Firstly, is it still okay to use assert to ensure denominators are not negative or should we ' fix' all negative denominators, for example in the extraction operator?

Also, I am in the process of adding documentation to the header file and I was wondering, do we provide any documentation about the constructor?

(https

Search entries or author

Friday

The answer to your first question was in the homework instructions (they are so lengthy I missed it the first time also): "Your simplify() function should also ensure that the denominator is never negative. If the denominator is negative, fix this by multiplying numerator and denominator by -1. Also, if the numerator is 0, the denominator should be set to 1."

To your second question yes I did and we should:)

Greg Maddox (https://santarosajc.instructure.com/courses/24402/users/61866)

(https://santarosajc.instructure.com/courses/24402/users/61866)

Friday

I'm still incredibly confused. If the simplify function doesn't turn a fraction into a mixed number, how does it simplify it? how do you reduce an improper fraction without converting it? what is the simplify function actually supposed to do?

Drew Rodrigues (https://santarosajc.instructure.com/courses/24402/users/59022)

(https://santarosajc.instructure.com/courses/24402/users/59022)

Friday

It reduces it based upon it's greatest common factor. Although it will still be a mixed number.

For example:

200/2 --> 100/1

150/5-->30/1

18/20-->9/10

You're just finding their greatest common factor, then dividing both numerator and denominator by it to reduce it. Hope this helps!

Greg Maddox (https://santarosajc.instructure.com/courses/24402/users/61866)

(https://santarosajc.instructure.com/courses/24402/users/61866)

Friday

ahhh, so its really a ratio reduction.

David Harden (https://santarosajc.instructure.com/courses/24402/users/60154)

(https://santarosajc.instructure.com/courses/24402/users/60154)

Search entries or author

Hi Gre

I don't know what a ratio reduction is. But, as Drew illustrated with his examples, simplifying fractions in mathematics has nothing to do with whether they are mixed numbers or not. Anyway...let me know if this hasn't cleared things up.

○ Greg Maddox (https://santarosajc.instructure.com/courses/24402/users/61866)
(https://santarosajc.instructure.com/courses/24402/users/61866)

Saturday

Yes it has, i don't know what i was saying yesterday, i was very tired working on it last night. Now i'm just having trouble with the actual algorithm, in that i mean finding the GCF. I've been spinning my wheels for several hours over this now, and i don't know why i can't figure it out. I can reduce fractions in my sleep, i just don't know how to test when the division isn't a whole number, as you would do when doing the prime factorization. maybe i'm just looking at it all wrong. Any thoughts?

○ David Harden (https://santarosajc.instructure.com/courses/24402/users/60154)
(https://santarosajc.instructure.com/courses/24402/users/60154)

Saturday

Use modulus to test when the division isn't a whole number.

If a % b is not 0, that means the division isn't a whole number.

Reply

○ Andrew Langwell (https://santarosajc.instructure.com/courses/24402/users/56027)
(https://santarosajc.instructure.com/courses/24402/users/56027)

Friday

Is it acceptable to have return statements within if statements in the simplify function? Should I find another way to write my simplify program that doesn't use return statements?

*Edited by **Andrew Langwell (https://santarosajc.instructure.com/courses/24402/users/56027)** on Sep 15 at 10:54am*

○ Drew Rodrigues (https://santarosajc.instructure.com/courses/24402/users/59022)
(https://santarosajc.instructure.com/courses/24402/users/59022)

Friday

What is your conditional statement checking for? I'm assuming you're using the return to just breakout of the function early in case it meets a certain criteria, such as having 1 in the denominator. Also, is your return statement

before
negati d
to retu

Search entries or author

return would only be purposeful for control flow, unless you don't have the return value of the function as void.

Andrew Langwell (https://santarosajc.instructure.com/courses/24402/users/56027)
(https://santarosajc.instructure.com/courses/24402/users/56027)

Saturday

Yes it was a quick fix for control flow to exit the loop after the numerator and denominator were divided by the GCF. I may look into finding a way to do without the return statements. However if some conditions are met and our fraction has been reduced, testing other conditional statements would only waste time. So return would allow us to exit once we've found the right conditions and reduced the fraction.

*Edited by **Andrew Langwell** (https://santarosajc.instructure.com/courses/24402/users/56027) on Sep 16 at 1:14pm*

David Harden (https://santarosajc.instructure.com/courses/24402/users/60154)
(https://santarosajc.instructure.com/courses/24402/users/60154)

Saturday

Assuming that your simplify() is void, you should restructure your loop to avoid the returns. This can be done without making anything less efficient. Regarding Drew's statement "I couldn't find a case where you would need to return early if you have your loop setup right," that's ***always*** true.

Andrew Langwell (https://santarosajc.instructure.com/courses/24402/users/56027)
(https://santarosajc.instructure.com/courses/24402/users/56027)

Saturday

Okay thanks. I'll try to simplify my code. The way my code is written, after my function has been divided by its GCF it will continue to see if the numerator and denominator can be divided by a divisor until the divisor is decremented to 2. Seems like a waste of time to do that. I'm having trouble finding a better way to write my loop to avoid this.

*Edited by **Andrew Langwell** (https://santarosajc.instructure.com/courses/24402/users/56027) on Sep 16 at 4:18pm*

David Harden (https://santarosajc.instructure.com/courses/24402/users/60154)
(https://santarosajc.instructure.com/courses/24402/users/60154)

Saturday

The way you should develop a loop is you should ***start*** by determining the conditions under which the loop should end. Those determine the condition of the loop. In this case, one of the conditions under which the loop should end is if the fraction has been divided by its GCF...

Craig Nicholas (https://santarosajc.instructure.com/courses/24402/users/118709)
(https://santarosajc.instructure.com/courses/24402/users/118709)

Saturday

I don't
sure t
then c

Search entries or author

to redo all that in some way to make sure it doesn't pointlessly enter the loop if one of those easy cases was encountered and already dealt with. Then I end up with some variables that are serving multiple purposes, carrying forward a task that depends on which conditionals were triggered. It's hard to see this as more comprehensible than doing early exits. Am I doing something wrong?

David Harden (https://santarosajc.instructure.com/courses/24402/users/60154)
(https://santarosajc.instructure.com/courses/24402/users/60154)

Sunday

One solution would be to use "else" to prevent the rest of the function from executing if one of those simple cases is met.

However, in most approaches to this problem, there's no need to make those into special cases anyway. They can be handled by the loop without making any adjustments. I'd have to see your code to say anything more specific. Feel free to email it to me and I'll take a look.

Andrew Langwell (https://santarosajc.instructure.com/courses/24402/users/56027)
(https://santarosajc.instructure.com/courses/24402/users/56027)

Sunday

Thanks for your help. I've found two solutions to avoid using the return statement and I think I've picked the best of the two. However my solution does use else if statements.

*Edited by **Andrew Langwell** (https://santarosajc.instructure.com/courses/24402/users/56027) on Sep 17 at 1:33pm*

Reply

(https://santarosajc.instructure.com/courses/24402/users/4932)
Kevin Drake
(https://santarosajc.instructure.com/courses/24402/users/4932)

Friday

In my client.pp the using namespace cs_Fraction is showing it not as a valid namespace item. Any ideas?

Kevin Drake (https://santarosajc.instructure.com/courses/24402/users/4932)
(https://santarosajc.instructure.com/courses/24402/users/4932)

Friday

I have already followed the directions in lesson 16.15

(https://santarosajc.instructure.com/courses/24402/users/102629)

Search entries or author

Were you sure to define it in the fraction.h file? Also look at your fraction.cpp file and make sure you implemented the namespace as in the example further in 16.16. I hope this helps.

(https://santarosajc.instructure.com/courses/24402/users/15956)
Greg Fischer
(https://santarosajc.instructure.com/courses/24402/users/15956)

Friday

namespace cs_Fraction{

...class and everything....

}//namespace cs_Fraction;
#endif /* fraction_h */

(https://santarosajc.instructure.com/courses/24402/users/60154)
David Harden
(https://santarosajc.instructure.com/courses/24402/users/60154)

Saturday

Kevin, still stuck on this? I think I'd need to see your code, if so.

(https://santarosajc.instructure.com/courses/24402/users/4932)
Kevin Drake
(https://santarosajc.instructure.com/courses/24402/users/4932)

Sunday

I figured it out. Thank you.

Reply

(https://santarosajc.instructure.com/courses/24402/users/63762)
Melissa Goh
(https://santarosajc.instructure.com/courses/24402/users/63762)

Friday

For simplify(), can I use another bool function to check if some value is an integer?

(https

Search entries or author

Friday

You are only dealing with integer values in simplify(). integer/integer or (integer, integer).

If you are asking how to deal with a mixed fraction with an integer at the start, the Extraction Operator >> should convert that into an improper fraction. So you are only dealing with an integer/integer in simplify().

(https://santarosajc.instructure.com/courses/24402/users/63762) Melissa Goh

(https://santarosajc.instructure.com/courses/24402/users/63762)

Friday

Thanks Greg! I was thinking it too mathematically.

Reply

(https://santarosajc.instructure.com/courses/24402/users/15956) Greg Fischer

(https://santarosajc.instructure.com/courses/24402/users/15956)

Friday

All my code is complete but in the header file (Fraction.h)

"Split the project up into three files: client file, implementation file, and header (specification) file."

1) What should be in the implementation file? I'm having trouble understanding how to split this all up.

2) Is there an example of how this all should look?

3) An example of this with documentation would also be helpful.

(https://santarosajc.instructure.com/courses/24402/users/68744) Eric Barnard

(https://santarosajc.instructure.com/courses/24402/users/68744)

Saturday

Go to your email Dave sent us on 09/07, it has access to the prior cs10 lessons. Lesson 15 is the first lesson on classes. Specfically, subsection 8 talks about multiple files.

(https://santarosajc.instructure.com/courses/24402/users/61866) Greg Maddox

(https://santarosajc.instructure.com/courses/24402/users/61866)

Saturday

Yeah, n
see w r
is just

Search entries or author

Reply

○ (https://santarosajc.instructure.com/courses/24402/users/118245)

Harry Doyle

(https://santarosajc.instructure.com/courses/24402/users/118245)

Saturday

Given I haven't taken college algebra in quite some time, so perhaps I am just way off base here and this is completely obvious, but I had some difficulty wrapping my head around -50/3 equaling -16+2/3. When I read -16+2/3's I think -48/3 + 2/3 = -46/3. The + sign does my mind in. I overcame it through code, as I converted everything to positive, got a 16 as an integer vaule by dividing, got the remainder by dividing again, and then just converted back to negative by multiplying by negative 1 (kept track with a boolean whether dealing with a negative value), and then put the string together accordingly which netted me out -16+2/3 , but in my mind this should have been -17+1/3 vs. -16+2/3.

○ (https://santarosajc.instructure.com/courses/24402/users/68744)

Eric Barnard

(https://santarosajc.instructure.com/courses/24402/users/68744)

Saturday

You're confusing the addition symbol '+', with the notation we just happen to be using.

We could just as easily use a hyphen, asterisk, or a caret.

In other words, Dave's instructions could have said to write a mixed fraction as:

-16^2/3
-16$2/3
-16&2/3

Or anything else he wanted, and it would mean the same thing at -16+2/3. The limitation here being we are using the command prompt/terminal, which tends to limit how we can output text.

Edit: Come to think of it, the ampersand may have been ideal to prevent this confusion, especially since we would tend to read it as "Sixteen and two thirds".

*Edited by **Eric Barnard** (https://santarosajc.instructure.com/courses/24402/users/68744) on Sep 16 at 2:46pm*

Reply

Search entries or author

○ (https://santarosajc.instructure.com/courses/24402/users/63762)

Melissa Goh

(https://santarosajc.instructure.com/courses/24402/users/63762)

Saturday

I think my Fraction class is written like how the majority of us are doing it. However, I'm calling simplify() 9 times (including the constructor). Would anybody care to share when should I be calling it and when I should not?

Edited by ***Melissa Goh*** *(https://santarosajc.instructure.com/courses/24402/users/63762) on Sep 16 at 3:32pm*

○ (https://santarosajc.instructure.com/courses/24402/users/118709)

Craig Nicholas

(https://santarosajc.instructure.com/courses/24402/users/118709)

Saturday

I take it the idea is to have the += type operators outsource "+" part to the "+" function, which presumably already has a call to simplify. Also if you dig into the logic of it you shouldn't need to simplify after using any of the ++ type operators. I don't know how important is it to minimize the uses of simplify() though.

Also it seems if you want to get ruthless about it you could rely on just two calls to simplify().

○ (https://santarosajc.instructure.com/courses/24402/users/56027)

Andrew Langwell

(https://santarosajc.instructure.com/courses/24402/users/56027)

Sunday

I would say that the most important place for simplify() is in the fraction constructor. That way you know any fractions going into your other functions are already in simplified form. Simplify in the fraction constructor covers what Dave hints at in the instructions: "Fractions should not be simply reduced upon output, they should be stored in reduced form at all times." The second most important place for simplify() in my code is in the extraction operator overload function. I'm not sure about the rest at the moment.

○ (https://santarosajc.instructure.com/courses/24402/users/63762)

Melissa Goh

(https://santarosajc.instructure.com/courses/24402/users/63762)

Tuesday

Thanks guys!

Reply

○ (https://santarosajc.instructure.com/courses/24402/users/56027)

Search entries or author

I'm struggling with the extraction operator overload function. Should the return type be fraction? I've managed to return the unsimplified fraction forms but at the moment I cant call the simplify function inside the operator overload function.

Greg Fischer (https://santarosajc.instructure.com/courses/24402/users/15956)

(https://santarosajc.instructure.com/courses/24402/users/15956)

Saturday

Return type should be an istream.

It's impossible it figure out without a good example.

Starting Out with Cpp 8th - page 849 is a basic example of how to do it.

Carolyn O'Rourke (https://santarosajc.instructure.com/courses/24402/users/102629)

(https://santarosajc.instructure.com/courses/24402/users/102629)

Saturday

Look at Dave's example in 16.16 how the overloaded operator + has a call within it to the private function

simplify() (answer.simplify();). If your simplify function is private within the class you must follow a similar format. Hope this helps

Ryan Martino (https://santarosajc.instructure.com/courses/24402/users/4456)

(https://santarosajc.instructure.com/courses/24402/users/4456)

Saturday

hey greg do you know what the books section was called where that example is found? its not on the same page in the 7th edition and I could really use something to push me in the right direction. you say the return type should be istream in all the examples we have done and assignment 3, I'm not sure if I'm supposed change it?

David Harden (https://santarosajc.instructure.com/courses/24402/users/60154)

(https://santarosajc.instructure.com/courses/24402/users/60154)

Sunday

Section 14.5, the subsection titled (surprisingly) "Overloading the << and >> Operators".

Or you can just follow the pattern of the example in lesson 16.10. That shows overloading the insertion operator. You just reverse everything for the extraction operator (istream instead of ostream, for example).

(https://santarosajc.instructure.com/courses/24402/users/4456)

Rya

(htt

Sun

Search entries or author

thanks dave!

Andrew Langwell (https://santarosajc.instructure.com/courses/24402/users/56027)

(https://santarosajc.instructure.com/courses/24402/users/56027)

Sunday

Thanks for the tips everyone. I don't have the textbook so I'll take a look at the example in the lesson.

Update: I was able to make my extraction operator overload function work properly. Thanks for all the help everyone. I still want to go over the function at some point to confirm my understanding of how it works.

Edited by ***Andrew Langwell*** *(https://santarosajc.instructure.com/courses/24402/users/56027) on Sep 17 at 1:36pm*

Reply

Ryan Martino (https://santarosajc.instructure.com/courses/24402/users/4456)

(https://santarosajc.instructure.com/courses/24402/users/4456)

Saturday

I was having a hard time getting the input file into Xcode because the video showing how uses text edit, I found a easy way around this by saving the file provided into Sublime Text 2 and saved it to desktop which makes it easy to find when you add file in the products folder in Xcode. Sublime Text 2 is free and very useful for programming, I use it to edit podfiles and such when working in the terminal. Might be a useful bit of software to download for future programming as well!

Reply

Kyle Stewart (https://santarosajc.instructure.com/courses/24402/users/85990)

(https://santarosajc.instructure.com/courses/24402/users/85990)

Saturday

When writing the Fractions to console, do we want the reader to assume -2 + 1/2 is

-4/2 + -1/2 = -5/2

or

-4/2 + 1/2 = -3/2

My gut sa

Search entries or author

What star

"A minus

Edited by ***Kyle Stewart (https://santarosajc.instructure.com/courses/24402/users/85990)*** *on Sep 16 at 5:44pm*

○ Carolyn O'Rourke (https://santarosajc.instructure.com/courses/24402/users/102629)

(https://santarosajc.instructure.com/courses/24402/users/102629)

Saturday

Nope the '+ ' is just a separator that you should use to know that there is more to come for the the fraction: -2+1/2 = -2 1/2 which you should store as -5/2 .

○ David Harden (https://santarosajc.instructure.com/courses/24402/users/60154)

(https://santarosajc.instructure.com/courses/24402/users/60154)

Saturday

Good choice not to go with your gut :)

Instead, go with what the assignment says: The '+' in mixed numbers does not mean add. It is simply a separator (to separate the integer part from the Fraction part of the number). So the Fraction "negative two and one-sixth" would be written as -2+1/6, even though -2 plus 1/6 is not what we mean.

Reply

○ (https://santarosajc.instructure.com/courses/24402/users/111918) Neal Siegrist

(https://santarosajc.instructure.com/courses/24402/users/111918)

Saturday

This assignment challenged with with the input from the file. It took me quite a bit of debugging to find out if the whole line was being read in or if it was a single character at a time and I needed to put it on a loop. Once I resolved this the rest of the assignment came fairly easy with a few minor tweaks. I feel like I am getting a better idea about classed and feel like I am able to use them more effectively than before especially with the operator overloading.

Reply

○ (https://santarosajc.instructure.com/courses/24402/users/112937) Lance Tran

(https://santarosajc.instructure.com/courses/24402/users/112937)

Sunday

What I th
get input
read on f

Search entries or author

wouldn't reduce, instead it would stay in "un"reduced form. The type fraction is an object which was instantiated or arranged in a respectful manner. The numerator and denominator should be in values of 0 and 1. This would make a new fraction object and also having the denominator to always be positive. I thought that was interesting.

Reply

[(https://santarosajc.instructure.com/courses/24402/users/94493)](https://santarosajc.instructure.com/courses/24402/users/94493)

Jeryn Inman

(https://santarosajc.instructure.com/courses/24402/users/94493)

Sunday

I had trouble with overloading the stream extraction operator, until Dave pointed out that it doesn't call the function character by character and that each case should store the entire fraction. Thanks Dave!

(https://santarosajc.instructure.com/courses/24402/users/84454)

Angela Mao

(https://santarosajc.instructure.com/courses/24402/users/84454)

Sunday

Thank you! I was having trouble with the stream extraction operator myself, but now that I know this it is starting to make more sense.

(https://santarosajc.instructure.com/courses/24402/users/4456)

Ryan Martino

(https://santarosajc.instructure.com/courses/24402/users/4456)

Yesterday

Im confused what you mean by this but I've been stuck on the extraction operator.. and if I'm supposed to be doing in >> , or temp >> ??

(https://santarosajc.instructure.com/courses/24402/users/60154)

David Harden

(https://santarosajc.instructure.com/courses/24402/users/60154)

Yesterday

The way that the extraction (>>) operator works is the place you are reading from goes first, then the operator, then the variable where you want the input to be stored. Like

cin >> hoursWorked;

Does that answer your question?

Reply

Search entries or author

○ (https://santarosajc.instructure.com/courses/24402/users/102629)

Carolyn O'Rourke (https://santarosajc.instructure.com/courses/24402/users/102629)

Sunday

I have a question regarding where we should put our comments for private class member functions. Should they be mentioned in the class invariant? Or should they just be above the function definition. I'm only asking about PRIVATE class functions. Can't seem to see this documented.

○ (https://santarosajc.instructure.com/courses/24402/users/102629)

Carolyn O'Rourke (https://santarosajc.instructure.com/courses/24402/users/102629)

Sunday

hello does anyone know?

○ (https://santarosajc.instructure.com/courses/24402/users/60154)

David Harden (https://santarosajc.instructure.com/courses/24402/users/60154)

Sunday

They should be above the function definition. Very last sentence in Style Convention 1D.2: Aside from the class invariant, the only comments you will need in your implementation file are comments on the implementation of complex functions, and comments on private functions (which do not get comments in the header file).

:)

○ (https://santarosajc.instructure.com/courses/24402/users/102629)

Carolyn O'Rourke (https://santarosajc.instructure.com/courses/24402/users/102629)

Sunday

Thanks

Reply

○ (https://santarosajc.instructure.com/courses/24402/users/76726)

David Novosartov (https://santarosajc.instructure.com/courses/24402/users/76726)

Monday

I really er
header fil
therefore

Search entries or author

Reply

○ (https://santarosajc.instructure.com/courses/24402/users/66384)
Julia Otten
(https://santarosajc.instructure.com/courses/24402/users/66384)
Monday

I found this assignment to be very interesting and not as complicated as I was afraid it might be for the most part. The simplify function didn't take me too long after I sketched out what I wanted it to do on paper and then wrote the code for it. I separated my code into three separate files and created the Fraction namespace first, and then started working on the assignment, for me it actually helps to have the code separate, I feel like it is more organized. Overloading the stream insertion operator wasn't too challenging for me, but I am still working on the stream extraction operator, hoping to finish up by tonight though!

Edited by ***Julia Otten*** *(https://santarosajc.instructure.com/courses/24402/users/66384) on Sep 18 at 1:01pm*

Reply

○ (https://santarosajc.instructure.com/courses/24402/users/73008)
Fred Straub
(https://santarosajc.instructure.com/courses/24402/users/73008)
Monday

It's been entertaining to apply the mathematics that I know to code. Most of this is something I can do without much thought, yet trying to tell my computer how to do it is amazingly challenging, and I am left impressed at our collective intelligence and the power of the human brain. I found that following the directions in order was the simplest way to complete this assignment, as in totality it initially appears overwhelming. Do working programmers (not students) jump to algorithms like Euclid's when in the "real world"?

Edited by ***Fred Straub*** *(https://santarosajc.instructure.com/courses/24402/users/73008) on Sep 18 at 2:19pm*

○ (https://santarosajc.instructure.com/courses/24402/users/68744)
Eric Barnard
(https://santarosajc.instructure.com/courses/24402/users/68744)
Monday

While not a professional programmer, or really even very competent one at this point, I do read enough forums and Stack Exchange to know that paid programmers' time is better spent simply using existing algorithms and libraries, instead of trying to write their own from scratch. Unless you're a brilliant mathematician who can come up with

somet
what v

Search entries or author

Sean Callahan (https://santarosajc.instructure.com/courses/24402/users/52648)

(https://santarosajc.instructure.com/courses/24402/users/52648)

Monday

After working on many open source programming projects, usually the first version of an algorithm is crude and verbose. It takes many iterations and contributions by many developers to optimize it and make it concise. However, like Eric mentioned, only a few brilliant minds can come up with a ground-breaking algorithm that solves the problem in the best way possible. Many mathematician and computer scientists spend their whole lives perfecting a solution to a particular problem.

David Harden (https://santarosajc.instructure.com/courses/24402/users/60154)

(https://santarosajc.instructure.com/courses/24402/users/60154)

Monday

Definitely, most "real" programmers will not be writing something like finding GCF from scratch. However, there will always be new problems to solve, so it's good practice. Also, you'll find that in this class mostly what we are doing is looking "under the hood". We'll be using a lot of techniques that are important to understand so that you are better able to use the tools (such as libraries) that are available.

Andrew Bieneman (https://santarosajc.instructure.com/courses/24402/users/99711)

(https://santarosajc.instructure.com/courses/24402/users/99711)

Monday

"It's been entertaining to apply the mathematics that I know to code"

I agree. I like math, but the higher level stuff gets a bit abstract and weird. I like how in programming classes like this we solve little logic/math problems, where the math isn't overly difficult, but a fun challenge. And the applications are practical.

Reply

David Holstedt (https://santarosajc.instructure.com/courses/24402/users/81230)

(https://santarosajc.instructure.com/courses/24402/users/81230)

Monday

I spent an hour trying to get the stream extraction operator declaration to work right and I'm pretty mad at C++ right now for the ambiguous and nonintuitive methods we have to use.

SHOW ALI

Search entries or author

1

(https

(https://santarosajc.instructure.com/courses/24402/users/81230)

Monday

I don't know if I going to get from the teacher before this is due, so can anyone see anything wrong with this declaration:

```
istream& operator>>(istream& in, Fraction& writeMe){

//code

}
```

*Edited by **David Holstedt (https://santarosajc.instructure.com/courses/24402/users/81230)** on Sep 18 at 8:32pm*

Alex Dewey (https://santarosajc.instructure.com/courses/24402/users/78834)

(https://santarosajc.instructure.com/courses/24402/users/78834)

Monday

I mean it's a bit late to type this now, but start from the basics and work your way up. Start with accepting the first values and slowly add more if statements and loops, eventually making your code into a monstrosity of garbage. I mean, that's what's working for me.

David Holstedt (https://santarosajc.instructure.com/courses/24402/users/81230)

(https://santarosajc.instructure.com/courses/24402/users/81230)

Monday

I have the if statements and stuff down. I just didn't want to show it here. What I was asking about was the syntax of the part outside the curly braces. I'm getting page long error messages on just that one part.

David Harden (https://santarosajc.instructure.com/courses/24402/users/60154)

(https://santarosajc.instructure.com/courses/24402/users/60154)

Monday

That header is correct. The problem must be somewhere else in your code. Is it declared as a friend in the prototype in the class declaration?

David Holstedt (https://santarosajc.instructure.com/courses/24402/users/81230)

(https://santarosajc.instructure.com/courses/24402/users/81230)

Monday

yes. I'll try that.

(https://santarosajc.instructure.com/courses/24402/users/81230)

Da

(htt

Mor

Search entries or author

I tried removing the friend declaration, but it still didn't work. I sent my code through email, If you could review it that would really help me out

(https://santarosajc.instructure.com/courses/24402/users/81230)

David Holstedt

(https://santarosajc.instructure.com/courses/24402/users/81230)

Monday

Here are the errors.

a4_1 errors.txt (https://santarosajc.instructure.com/files/962140/download?download_frd=1&verifier=Q5NiQoLCJe9FBglCgLtrJsWx9Nix2YAHn8AD3s0b)

(https://santarosajc.instructure.com/courses/24402/users/74745)

James O'Hara

(https://santarosajc.instructure.com/courses/24402/users/74745)

Monday

It shouldn't be a member function. It should be a friend. A little earlier Dave asked about that, and I think you thought he meant it shouldn't be a friend, but he meant it should.

From the attachment: Fraction::operator>>(std::istream&, Fraction&)í must take exactly one argument [because, I guess, it assumes the Fraction argument is the calling object? But if it's not a member function (no Fraction:: qualifier), then it can take two arguments].

Reply

(https://santarosajc.instructure.com/courses/24402/users/78465)

Sutter Laird

(https://santarosajc.instructure.com/courses/24402/users/78465)

Monday

Overloading the stream extraction/insertion and postfix/prefix increment/decrement operators were the biggest challenge in this assignment for me, though finding an alternative to Euclid's algorithm for reducing fractions was also an interesting bit of experimentation. Overall I enjoyed this assignment and the trial-and-error learning method it featured.

Reply

(https://santarosajc.instructure.com/courses/24402/users/111342)

Search entries or author

Much of the time of this assignment came from ensuring everything was properly documented. Really happy with the algorithm generated for the 'simplify' function but it took a bit of trial and error to create. Overall, makes you really appreciate the effort that goes into the documentation for various libraries that you may find online.

Reply

○ (https://santarosajc.instructure.com/courses/24402/users/63243)

Nathan Cortez

(https://santarosajc.instructure.com/courses/24402/users/63243)

Tuesday

Pretty interesting! Getting everything up and running was pretty satisfying. I actually did an extra credit assignment on the last course that involved reducing fractions and I was able to simply take that formula and modify it to fit this assignment.

My biggest problem was outputting the mixed fractions and that took a bit of work until I found a clean and simple way of doing it. It actually involved using absolute values as it really helped shorten down the code quite a bit.

Reply

○ (https://santarosajc.instructure.com/courses/24402/users/110663)

Alan Becker

(https://santarosajc.instructure.com/courses/24402/users/110663)

Tuesday

The most satisfying part of this assignment was simplifying the simplify() function (hah!) that I'd created for CS10 so that it loops a lot fewer times in a lot of cases, especially very small fractions. Documentation took a pretty long time for this one, but that's probably because I used it as an opportunity to revise a decent chunk of my code. Unexpectedly, I ended up really appreciating the perspective that documenting the more complex functions gave me, as it caused me to think of ways to simplify a few of them quite a bit.

Reply

○ (https://santarosajc.instructure.com/courses/24402/users/81540)

Search entries or author

I found the discussion board to be incredibly helpful with this assignment. In the beginning I was staring at the instructions, trying to make sense of what was being asked (especially with simplify). Just scrolling through some of comments put me on the right track and I was able to finish the coding on my own.

Reply

○ (https://santarosajc.instructure.com/courses/24402/users/3920) Alexis Ordonez

(https://santarosajc.instructure.com/courses/24402/users/3920)

Yesterday

Honestly, I love and hate assignments like this. I love the arithmetic, I was a math major prior to this, but I hate trying to make a client file work when I didn't write it. I fou [illegible] ng v [illegible] s [illegible] doing a loop t [illegible] g both the numerator and denominator will be while numbers. If they could both be divided by the same number and both produce a whole number, they would be simplified. The loop continues to check each value between 2, since dividing by one is redundant, and then checks 3, 4, 5, 6, etc. until it reaches a max number determined by the numerator and the denominator. The input was trickier for me to figure out just because of the + between the numbers. I found doing a value swap would handle both the whole number value and include the numerator and denominator. The output also needed some updating but that was more common sense and having the output determine the best form to print the fraction depending on the values of both the numerator and denominator, such as if the denominator equals 1, just output the numerator as a whole number, or the numerator equaling 0, simply output the numerator since (assuming the denominator is not 0) no matter what the denominator is, its value will be 0. Again, fun with the arithmetic, not so fun with all the trying to make the puzzle pieces fit.

Unread

✓ Subscribed

Reply

○ (https://santarosajc.instructure.com/courses/24402/users/66085) Jeremy King

(https://santarosajc.instructure.com/courses/24402/users/66085)

Yesterday

I definitely think I will make some more time to work on upcoming assignments.

○ (https://santarosajc.instructure.com/courses/24402/users/57399) Sam Ha

(https://santarosajc.instructure.com/courses/24402/users/57399)

Yesterday

the ist

Search entries or author

Reply

○ (https://santarosajc.instructure.com/courses/24402/users/50114)

Anthony Chui

(https://santarosajc.instructure.com/courses/24402/users/50114)

Yesterday

Must the arithmetic and comparison functions be written as friend functions?

Reply

○ (https://santarosajc.instructure.com/courses/24402/users/64934)

Hugh Lin

(https://santarosajc.instructure.com/courses/24402/users/64934)

Yesterday

I'm getting the assertion failed error at line 45 in the client program where the code is "assert(in)". Any help?

Reply

○ (https://santarosajc.instructure.com/courses/24402/users/11657)

Tommy Streb

(https://santarosajc.instructure.com/courses/24402/users/11657)

Yesterday

I feel like I'm making things more complicated than they need to be, but the only thing confusing me is the istream overload, because I don't understand where I'm returning the value being input to. I never learned anything about peek() or ignore() in CS10 along with a lot of other built in functions, so I'm pretty lost with that, but I'll keep looking.

Reply