CS 460 – Exercise 2 - FAQ

Question: The columns in the email I received are completely messes up. What do I do?

```
<     3   $12.34   -$12.34   -$24.68   Insufficient amount tendered
<     4    $0.00    $12.34    $12.34    4    1       1       1       1
<     5  -$12.34    $12.34    $24.68    3    1    1       1    4           1
<     6   $12.34    $34.56    $22.22    2       2           2           1
<     7    $0.00     $0.00     $0.00
<     8    $0.00     $0.00     $0.00
< Totals     $2.90    $79.31    $76.41   11    3    4    1    3   14        2    2
\ No newline at end of file
---
>     6   $12.34   -$12.34   -$24.68   Insufficient amount tendered
>     7  -$12.34   -$12.34     $0.00
>     8   $12.34    $12.34     $0.00
>     9   $34.56   $123.45    $88.89    4       1    1    1    3    1    1    1    1
>    10    $0.00     $0.00     $0.00
> Totals    $52.53   $166.31   $113.78    8    2    1    2    2   12    2    2    1    1
```

Answer: The font used by your mail client is not a fixed width font. Copy the lines to a document and change the font to a fixed width font like Courier New or Menlo.

Question: I am getting an extra transaction at the end of the input with 0.00 as both the purchase and the tendered. There is not a line in the input file with 0.00 0.00 on it.

Answer: It is probably related to how you are looking for the end of data. Emacs and Vim use different formats and the ios::eof() method does not return consistent results.

Question: I received the email back from you but it said it was unable to compile and I have absolutely zero idea why it won't compile. It compiles absolutely just fine on blue for me but in the email it says it cannot stat Ex2.exe, even though it is in my make file with the make name of Ex2.exe and when I run 'make Ex2.exe' it works just fine.

Answer: The first lines of your makefile should be:

```
     Ex2.exe : Exercise2.o money.o
          g++ –o Ex2.exe Exercise2.o money.o
```

Question: When I was calculating the sum of all the purchases, I got the total 1393.01, however in the exercise 2 pdf it shows the total sum as 1387.24. I checked on two different calculators and they also got 1393.01. Maybe there is something I am missing. Any clarification will be appreciated.

Answer: Transactions where the tendered is less than the purchase should display "`Insufficient amount tendered`" and neither the purchase amount nor the tendered amount should be included in the totals.

Question: I'm having a hard time getting the columns to align in the output. I'm trying to use setw() with the widths in the spec but cannot get the data to align for the life of me and Google isn't helping much either. What is your advice for getting the data to align properly? Should we be using setw()?

Answer: The best way to do this is to write the formatted money string to a stringstream – then write the string stored in the stringstream to the passed output stream.

Question: I have a question regarding Exercise 2. When we are overloading the modulo operator, how should we be interpreting the value of the parameter F? Is this a value that is in dollars or cents?

Answer: F should be an integer. The result is the remainder when the money object is divided by the integer: $1.25 % 6 ➔ $0.05

Question: When documenting the methods/constructors in the money.h file with the doxygen-style comments, do you want us to copy and paste those exact same comments into the money.cpp file? It feels redundant to document both the same way, especially when the interface is supposed to be the thing "users" of the API would look at, the whole point of having an interface.

Answer: The documentation if the .h file is for the user of the class and should explain how to use each function. The documentation in the .cpp file is for a programmer who might be modifying the class and needs to know the nitty-gritty details of how it works and why.

Question: If we covered this in class I must've forgot, but how are we supposed to keep record of a negative currency? Since our currency array is unsigned and we cannot add a getter for a potential negative value to our public, I'm struggling to see how we would handle having negative currency. This is especially confusing to me because of this test case in the example testcases:
2 $5.77 $5.00 -$0.77 Insufficient amount tendered
Would we just put the -$0.77 output in our subtraction overload and return an object with 0 dollars and 0 cents?

Answer: Yes – the value stored in the object should be -0.77; the currency array should actually be empty (size = 0, currency = NULL).

Question: I also noticed that your totals output (at the very bottom) lists a set of denominations that is not optimized is that the correct output?

Answer: Yes, the totals at the bottom of the output are the totals for each currency denomination.

Question: I understand, (correct me if I'm wrong), that if our dynamic array contains [101] it means we have 1 penny, 0 nickels, and 1 dime (assuming we are going in ascending order). So, if we wanted to produce our currency array for $100 it would look something like this? [0,0,0,0,0,0,0,0,0,0,1]

Answer: Since your currency array is private, you can store the information in either ascending or descending order.

Question: How exactly should we update our currency array? For example, if we initialized an object that contained $6.97, where would our calculations, (to find how many and what kind of denominations are needed), go? I understand we could initially do the calculations in that constructor, however, if we were to add to that same object we would need to update our currency array. **Would we be able to have a helper function that aids in doing that?**

Answer: Yes, a private helper function is a great idea!

Question: I see we have a set currency function that is supposed to set our currency array. Can we assume that that calculation is done apart from the class and that the vector passed already contains the updated/optimized denominations count?

Answer: The vector that is passed on set currency will contain denomination counts. They may not be optimal.